Gaussian Control Barrier Functions: Non-Parametric Paradigm to Safety

MOUHYEMEN A. KHAN1, TATSUYA IBUKI2, (Member, IEEE), AND ABHIJIT CHATTERJEE1, (Fellow, IEEE)

1School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
2Department of Electronics and Bioinformatics, School of Science and Technology, Meiji University, Kawasaki, Kanagawa 214-8571, Japan

Corresponding author: Mouhyemen A. Khan (mouhyemen.khan@gatech.edu)

This work was supported by the U.S. National Science Foundation under Grant S&AS:1723997 and Grant CCF 2128419.

ABSTRACT
Inspired by the success of control barrier functions (CBFs) in addressing safety, and the rise of data-driven techniques for modeling functions, we propose a non-parametric approach for online synthesis of CBFs using Gaussian Processes (GPs). A dynamical system is defined to be safe if a subset of its states remains within the prescribed set, also called the safe set. CBFs achieve safety by designing a candidate function a priori. However, designing such a function can be challenging. Consider designing a CBF in a disaster recovery scenario where safe and navigable regions need to be determined. The decision boundary for safety here is unknown and cannot be designed a priori. Moreover, CBFs employ a parametric design approach and cannot handle arbitrary changes to the safe set in practice. In our approach, we work with safety samples to construct the CBF online by assuming a flexible GP prior on these samples, and term our formulation as a Gaussian CBF. GPs have favorable properties such as analytical tractability and robust uncertainty estimation. This allows realizing the posterior with high safety guarantees while also computing associated partial derivatives analytically for safe control. Moreover, Gaussian CBFs can change the safe set arbitrarily based on sampled data, thus allowing non-convex safe sets. We validated experimentally on a quadrotor by demonstrating safe control for 1) arbitrary safe sets, 2) collision avoidance with online safe set synthesis, 3) and juxtaposed Gaussian CBFs with CBFs in the presence of noisy states. The experiment video link is: https://youtu.be/HX6uokvCiGk.

INDEX TERMS
Control barrier functions, Gaussian processes, non-parametric, safety-critical control.

I. INTRODUCTION
With the rise of autonomous systems, assuring their safety is of paramount importance. For instance, an autonomous drone should not crash during its mission, or a self-driving vehicle should not collide with other vehicles. Formulating constraints for these applications to ensure safety is a difficult task. These intelligent systems often use data-driven solutions for learning and adapting online. However, it is unclear how to principally and efficiently encode data into the paradigm of safety. A popular approach to ensuring safety of dynamical systems leverages set theoretic ideas. To be more specific, the theory of controlled set invariance is employed where a system is defined to be safe if (a subset of) its states remain within a prescribed set [1]. This forms the basis of control barrier functions (CBFs) which have been successfully demonstrated on many safety-critical applications [2], [3], [4]. To incorporate safety using CBFs, we need two items: a candidate function or certificate satisfying the required relative degree and a nominal model of the system dynamics. The candidate function defines a superlevel set in which the system remains forward invariant subject to certain constraints. Traditionally, these safety functions have been hand-designed. However, depending on the application, designing such a candidate CBF is not straightforward in many practical settings. To highlight the difficulty of a hand-designed solution, consider the example of a disaster recovery scenario. Designing a safety function manually for this scenario...
safety samples, which encode a safety metric of interest, with many of these concerns. To this end, we propose working sampled data in a principle and efficient manner will alleviate we believe designing the safety candidate function based on this can compromise system safety if designed incorrectly. CBFs achieve safe control using barrier certificates [5], [6]. The system’s safety is encoded using these safety barrier certificates (or safe sets) with the aid of a smooth function satisfying certain properties. These functions can then be combined with quadratic programs (QPs) to achieve safety constrained control [2], [3], [7], [8]. Certificates based on Lyapunov and barrier functions were combined to demonstrate stable and safe constrained control [9], [10]. These approaches incorporate deterministic CBFs defined a priori without incorporating any form of data for altering the CBF.

Learning based methods for addressing safety have been investigated previously. For an uncertain nonlinear system, the region of attraction is learned by using non-parametric GPs and Bayesian optimization (BO) to estimate and expand the safe set in [11]. BO was also used in safety-critical systems such as quadroxed, snake, and quadrotor for improving system performance while ensuring safety [12], [13], [14]. Unfortunately, the high run-time complexity of BO limits its applicability to evolving tasks or changing environmental conditions. For systems with polynomial dynamics, an optimization routine can be set up as a convex semi-definite problem using sum-of-squares (SoS) technique to search for a valid safety certificate [15], [16]. However, SoS methods also scale poorly with high dimensions, similar to BO, and are limited to polynomial system dynamics.

In the context of CBFs, data-driven techniques are actively pursued. Support vector machines (SVMs) were used in [17] to parameterize CBFs with the help of sensor measurements. Carefully designed weights are required in [17] for the SVM classifier to work and the study is confined to simulation results. Data in the form of expert demonstrations was used in [18], [19] to generate CBFs. However, in many applications, having access to expert demonstrations is not always feasible. While both the papers [17] and [18] empirically verify their findings, neither provide hardware experimental validation of their methods. Adaptive CBFs were formulated to handle time-varying control bounds and noise in the system dynamics [20]. CBFs have been combined with model predictive control methods for safe motion and path planning [21], [22]. For stochastic dynamical systems, stochastic CBFs are developed [23]. These approaches use parametric CBFs while accounting for uncertainty in the system dynamics. The authors in [24] use a value-function approach by combining Hamilton-Jacobi (HJ) with CBFs to maximize the safe set using viability kernels. However, the HJ based CBF is limited to low-dimensional systems.

Neural certificates, which use neural networks to construct safety barrier certificates, have been demonstrated in [25], [26], [27], and [28]. These neural certificates provide a data-driven approach to learning-based controllers and provide formal proofs of correctness. A second-order cone program was formulated in [29], with GPs used for modeling the control input and dynamic model uncertainty learned in an episodic manner. However, all these studies have been confined to simulation experiments and are limited to offline training which limits their applicability in many practical online settings. The work in [29] was limited to a particular type of kernel to satisfy affineness properties. We differ from [29] in modeling the safety candidate function with GPs instead of the underlying system dynamics and are not limited to one type of kernel parameterization. In our previous work [30], safety uncertainty was introduced in CBFs by augmenting the GP posterior variance with an existing CBF using data. We experimentally validated augmentation of safety uncertainty to a given CBF in hardware. However, this required a parametric CBF as the underlying safety function. Moreover, the safe set expansion in [30] was limited only to convex safe sets. In this research, we consider a fully non-parametric formulation for synthesizing the safety function without requiring any parametric CBF candidate function.

We also provide theoretical guarantees in this study. Our main contributions are the following.

1) We present a novel approach for synthesizing CBFs in a data-driven non-parametric manner using GPs. This is achieved using safety samples as opposed to the prevailing use of CBFs which requires a function.

2) We construct Gaussian CBFs to design safe sets based on the data. These sets are not confined to convex safe sets. GPs provide favorable properties such as analytical tractability and uncertainty estimation which are key enablers in finding closed-form safety function and associated Lie derivatives with high guarantees.

3) We formulate Gaussian CBFs for safe control in the presence of noise for both the safety samples (observations to GPs) and the system states (inputs to the GPs).
4) We validate Gaussian CBFs in hardware using a quadrotor for three case studies: (i) safe control for fixed but arbitrary safe sets, (ii) online obstacle avoidance with an evolving safe set, and (iii) juxtaposing Gaussian CBFs with regular CBFs for safe control in the presence of noisy system states.

To the best of our knowledge, we believe this is the first work that fully synthesizes a CBF in a non-parametric data-driven manner online using GPs and validate all findings in hardware. The outline of the paper is as follows. Mathematical preliminaries are covered in Section II with the problem statement discussed in Section III. We present our proposed methodology in IV. The quadrotor application testcase is explained in Section V. We discuss experimental results in Section VI followed by conclusion in Section VII.

II. BACKGROUND PRELIMINARIES
We tackle the problem of encoding safety for a dynamical system using a data-driven approach. We leverage theoretical properties of CBFs and GPs to encode safety in a probabilistic manner. We first review key results on CBFs and GPs and refer the reader to [2], [31], and [32] for more details.

A. CONTROL BARRIER FUNCTION
Consider a general control affine dynamical system,

\[ \dot{x} = f(x) + g(x)u, \]  

where \( x(t) \in \mathbb{R}^n \) is the state and \( u(t) \in \mathbb{R}^m \) is the control input. The drift vector field, \( f : \mathbb{R}^n \to \mathbb{R}^n \), and control matrix field, \( g : \mathbb{R}^n \to \mathbb{R}^{n \times m} \), are assumed to be locally Lipschitz continuous. For reasons of brevity, we omit the implicit dependence on time for the state and control input unless otherwise stated. Let safety for (1) be encoded as the superlevel set \( S \) of a smooth function \( h : \mathbb{R}^n \to \mathbb{R} \) as follows,

\[ S = \{ x \in \mathbb{R}^n \mid h(x) \geq 0 \}. \]  

Definition 1 (Control Barrier Function [2]): The function \( h(x) : \mathbb{R}^n \to \mathbb{R} \) is defined as a control barrier function (CBF), if there exists an extended class-\( \kappa \) function \( \alpha (\alpha(0) = 0 \text{ and strictly increasing}) \) such that for any \( x \in S \),

\[ \sup_{u \in \mathbb{R}^m} L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0, \]  

where \( L_f h(x) = \frac{\partial h}{\partial x} f(x) \) and \( L_g h(x) = \frac{\partial h}{\partial x} g(x) \) are the Lie derivatives of \( h(x) \) along \( f(x) \) and \( g(x) \) respectively.

Theorem 1 (Safety Condition [2]): Given a system (1), with safe set \( S \subset \mathbb{R}^n \) and a smooth CBF \( h(x) : \mathbb{R}^n \to \mathbb{R} \) (3), any Lipschitz continuous controller \( u \in \mathbb{R}^m \), chosen from \( K_{cbf} = \{ u \in \mathbb{R}^m \mid L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0 \text{ for any } x \in \mathbb{R}^n \} \), renders the set \( S \) forward invariant for (1).

As seen from Theorem 1, CBFs are limited to systems with relative degree one, \( \rho = 1 \). For systems with \( \rho > 1 \), we look at an extension of CBFs called Exponential CBFs [31, 33].

Definition 2 (Exponential Control Barrier Function [31]): The smooth function \( h(x) : \mathbb{R}^n \to \mathbb{R} \), with relative degree \( \rho \), is defined as an exponential control barrier function (ECBF), if there exists \( K \in \mathbb{R}^\rho \) such that for any \( x \in \mathbb{R}^n \),

\[ \sup_{u \in \mathbb{R}^m} L_f^\rho h(x) + L_g^\rho h(x)u + \rho ! H \geq 0, \]

where \( H = [h(x), L_f h(x), \ldots, L_f^{\rho-1} h(x)]^\top \in \mathbb{R}^\rho \) is the Lie derivative vector for \( h(x) \), and \( \rho ! = [k_0, k_1, \ldots, k_{\rho-1}]^\top \in \mathbb{R}^\rho \) is the coefficient gain vector for \( H \). \( K \) can be determined using linear control methods such as pole placement. We refer the reader to [31] for proofs of ECBF forward invariance.

B. GAUSSIAN PROCESS REGRESSION
GPs are a popular choice in machine learning for nonparametric regression which rely on kernels. Kernels furnish a notion of similarity between pairs of input points, \( x_i, x_j \in \mathbb{R}^n \). However, any arbitrary function of input pairs will not constitute a valid kernel. To be a valid kernel, it should satisfy positive semidefiniteness, see [32]. A popular choice of the kernel function is the squared exponential (SE) kernel,

\[ k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^\top L^{-2}(x_i - x_j)}{2}\right) + \delta_{ij}\sigma_y^2, \]

where \( \delta_{ij} = 1 \) if \( i = j \) and 0 otherwise, \( L \in \mathbb{R}^{n \times n} \) is the characteristic length scale, with \( L = \text{diag}(l) \in \mathbb{R}^{n \times n} \). The signal scale and observation noise are given by \( \sigma_f^2 \in \mathbb{R} \) and \( \sigma_y^2 \in \mathbb{R} \) respectively. Together, these free parameters constitute the SE kernel’s hyperparameters, \( \Theta = [L, \sigma_f^2, \sigma_y^2] \).

We are interested in constructing a safety function for which we assume to have noisy scalar observations. Given a set of \( N \) data points, with input vectors \( x \in \mathbb{R}^n \), and scalar targets \( y \in \mathbb{R} \), we compose the dataset \( D_N = \{ x_N, y_N \} \), where \( x_N = \{ x_i \}_{i=1}^N \) and \( y_N = \{ y_i \}_{i=1}^N \). GPs can compute the posterior mean and variance for an arbitrary deterministic query point \( x_q \in \mathbb{R}^n \), by conditioning on previous measurements. We will later investigate how to handle the case when the query point \( x_q \) is noisy. The posterior mean \( \mu \in \mathbb{R} \) and variance \( \sigma^2 \in \mathbb{R} \) are given by [32],

\[ \mu(x_q) = k(x_q) \top K^{-1} y_N, \]

\[ \sigma^2(x_q) = k(x_q, x_q) - k(x_q) \top K^{-1} k(x_q), \]

where \( k(x_q) = [k(x_1, x_q), \ldots, k(x_N, x_q)]^\top \in \mathbb{R}^N \) is the covariance vector between \( x_N \) and \( x_q \), \( K \in \mathbb{R}^{N \times N} \), with entries \( K_{i,j} = k(x_i, x_j) \), \( i, j \in \{ 1, \ldots, N \} \), is the covariance matrix between pairs of input points in \( x_N \), and \( k(x_q, x_q) \in \mathbb{R} \) is the prior covariance. The SE kernel is infinitely differentiable, and hence, it is infinitely mean-square (MS) differentiable. This allows for a flexible parameterization of the safety function and its Lie derivatives. We use the SE kernel to develop Gaussian CBFs while noting that any valid kernel can be used.

III. PROBLEM STATEMENT
Consider a control affine system (1) is given, with access to its states \( x \), and scalar noisy observations \( y \), that represents a metric for safety. The metric for safety cannot be generalized
and therefore is very problem dependent. A distance sensor’s readings for obstacle avoidance can be used as a metric for safety or a temperature sensor’s readings for determining thermally acceptable regions to traverse. In a similar vein, a LIDAR scan creating 3D point cloud information can be used to detect environmental hazards or a computer vision algorithm providing the decision boundary for safe regions of interest. In all these examples, we can easily sample from the data based on domain knowledge to construct a target metric for safety. This provides us with the means to construct a valid safety certificate using the data as opposed to hand-designing a safety function which could be limited and requires manual effort along with good domain knowledge intuition.

Remark 1: We assume there is a high-level planner or observer, e.g., sensors or computer vision algorithms, providing the necessary data observations. We acknowledge some feature engineering or data sampling may be involved which is very common in practice. These observations represent the safety sample candidates in our problem setting.

Our objective is to synthesize a safety function $h_{\text{gp}}(x)$ in a non-parametric manner from measurements of the system states and safety samples or observations online and ensure that (1) remains safe. Data-based methods are ultimately approximations and hence, it is desirable to account for any uncertainty in the estimation of the safety function. This leads to the following candidate function,

$$h_{\text{gp}}(x(t)) := h_b(x(t); \Theta) - h_u(x(t); \Theta).$$

The system’s overall safety is given by $h_{\text{gp}}(x)$ which has two components; a belief in safety given by $h_b(x)$ and an associated uncertainty given by $h_u(x)$. Intuitively, the safety belief represents our best estimation of system safety and safety uncertainty represents the uncertainty in the estimation. Ideally, if there is no uncertainty, then the safety belief will perfectly match the overall final safety. Additionally, there are hyperparameters $\Theta$ that can alter the relative notion of safety belief and uncertainty.

Problem 1: Given system (1) and online (noisy) measurements of the state $x$, synthesize $h_{\text{gp}}(x)$ with a safety belief and associated uncertainty, conditioned on past states and observations in the dataset given by: $D_N = \{x_N, y_N\}$, where $x_N = \{x_i\}_{i=1}^N$ and $y_N = \{y_i\}_{i=1}^N$, such that system (1) is safe.

To ensure the system remains safe, we need to rectify a given nominal control input $u_{\text{nom}}$ to its rectified form $u_{\text{rect}}$ which is then applied to (1). This is done by making sure the Lie derivatives of the corresponding candidate safety function satisfies the inequality (3).

Problem 2: Given system (1), synthesized $h_{\text{gp}}(x)$ with safe set $S$, and a nominal control input $u_{\text{nom}}$, design the rectified control input $u_{\text{rect}}$ such that system (1) is safe.

A non-parametric approach is adopted where the data is fully exploited to construct the safe sets and safety function hypothesis. This compounds the problem of computing the Lie derivatives since time derivatives are computed on $h_{\text{gp}}(x)$ in order to satisfy the forward invariance properties for CBFs. It is an ill-posed problem to compute the time derivative of an unknown entity i.e., the system’s safety belief and uncertainty, without making prior assumptions. For addressing these challenges, a kernel representation is used for constructing the safety belief and uncertainty.

IV. PROPOSED METHODOLOGY

In this section, we present our proposed approach, where GPs are used for synthesizing the safety function. A key advantage of GPs over other models such as neural networks, radial basis functions or polynomial chaos, lies in its bayesian non-parametric design. By allowing a flexible prior over functions, GPs give a probabilistic workflow that gives robust posterior estimates in analytical form. This enables a flexible realization for our safety function as well as computing the associated Lie derivatives. The resulting architecture for our framework is shown in Figure 2.

A. GAUSSIAN CONTROL BARRIER FUNCTION

A GP prior is placed on the desired candidate safety function, $h_{\text{gp}}(x) \sim GP(0, k(x, x'))$. By using the GP prior, we fully specify the candidate safety function unlike our previous work in [30]. Only the GP posterior variance was used in [30] to handle safety uncertainty for a given deterministic CBF. Note that the new formulation is a far more flexible realization since the data is used for informing both the safety belief and associated uncertainty. We operate under the following standard assumptions for GPs.

Assumption 1: Each observation $y_i$ is corrupted with Gaussian noise, $y_i \sim N(p_i, \sigma_y)$, where $p_i$ is the noise-free safety sample and $\sigma_y$ is the observation noise variance.

Assumption 2: Training input states $X_N$ in the dataset $D_N$ are noise-free.

We consider the case of getting noisy measurements for the safety samples. This is a very realistic assumption, since in practice these safety samples are captured from noisy sensory measurements. The input training data is considered to be noise-free, however, which is a common practice in machine learning. We first consider the case where the query point $x_q$ is deterministic, i.e., noise-free. Later, we look at the case when the input query point is also noisy, see Section IV-D.

Assumption 3: The safe set is nonempty with at least one datapoint, the initial state $x(0)$ and associated safety value $h_{\text{gp}}(x(0)) \in \mathbb{R}_{\geq 0}$, to synthesize $h_{\text{gp}}$.

We assume that the system begins in an initial compact safe set. Safety for $h_{\text{gp}}$ is encoded as,

$$S = \{x \in \mathbb{R}^n \mid h_{\text{gp}}(x) \geq 0\}.$$  \hspace{1cm} (8)

\[ \partial S = \{x \in \mathbb{R}^n \mid h_{\text{gp}}(x) = 0\}. \]  \hspace{1cm} (9)

A noise-free query state $x_q \in \mathbb{R}^n$ with a noisy safety sample $y \in \mathbb{R}$ is sampled if,

$$\|x(t) - x_q\| \geq \tau, \quad i = \{1, \cdots, N\},$$  \hspace{1cm} (10)
where \( r \in \mathbb{R} \) is the sampling distance between any two input states. This avoids dense sampling of the states, resulting in computational tractability. Moreover, if the samples collected are too close together, this may give rise to an ill-conditioned covariance matrix [32].

**Definition 3 (Gaussian Control Barrier Function):** A function \( h_{gp}(x) : \mathbb{R}^n \rightarrow \mathbb{R} \) is defined as a Gaussian CBF for (1), if \( h_{gp}(x) \sim GP(0, k(x, x)) \) is a Gaussian process, with an infinitely mean-square differentiable positive definite kernel, \( k(x, x) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \), and if \( \exists \) an extended class-\( \kappa \) function \( a \) such that for any \( x \in \mathbb{R}^n \),

\[
\sup_{u \in \mathbb{R}^m} L_f h_{gp}(x) + L_u h_{gp}(x) u + a(h_{gp}(x)) \geq 0. \tag{11}
\]

**Remark 2:** The Gaussian CBF above has attractive properties. A GP prior is placed on the safety candidate function, giving rise to a non-parametric functionality. Thus, the data is used to fully realize the safety function a posteriori. As more data is collected, the overall safety encoded by \( h_{gp}(x) \) changes. Moreover, it has an analytical form for both the safety belief and uncertainty. This enables computing Lie derivatives of \( h_{gp}(x) \) in closed-form.

The Lie derivatives in (11) require taking partial derivatives of \( h_{gp} \) with respect to \( x \) which we will discuss later in Section IV-B. We propose the Gaussian CBF \( h_{gp}(x) \) that incorporates safety belief and uncertainty online using the GP posterior mean (5) and variance (6) as follows,

\[
h_{gp}(x) := \mu(x) - \sigma^2(x) = \mathbf{k}(x)^\top \mathbf{K}^{-1} \mathbf{y}_N - (\mathbf{k}(x, x) - \mathbf{k}(x)^\top \mathbf{K}^{-1} \mathbf{k}(x)). \tag{12}
\]

The GP posterior mean represents the belief we have regarding safety whereas the GP posterior variance accounts for safety uncertainty. We require the following theorem to discuss forward invariance properties for the set \( S \) in (8)-(9).

**Theorem 2 (Sample Path Differentiability [34]):** A Gaussian process with an isotropic correlation function that can be expressed in the Schoenberg representation [35], has \( M \)-th-order mean-square partial derivatives if \( M \) moments of the length-scale parameter, \( l \), are finite.

We first consider an unforced dynamical system given by \( \dot{x} = f(x) \), where \( u(t) = 0, \forall t \geq 0 \). In this case, the Gaussian CBF will simply be considered as a Gaussian barrier function, since the control input does not appear.

**Proposition 1:** Given a system \( \dot{x} = f(x) \) with a nonempty safe set \( S \) as defined by (8) for a Gaussian process \( h_{gp} \), if \( h_{gp} \) is a Gaussian barrier function defined on the set \( S \), then \( S \) is forward invariant.

**Proof:** First, we observe that \( h_{gp} \) uses an infinitely MS differentiable kernel. Hence, \( h_{gp} \) is also infinitely MS differentiable with respect to \( x \) due to Theorem 2 since the length-scale has infinitely many moments. Since \( h_{gp} \) is a Gaussian CBF and infinitely MS differentiable, then the inequality \( L_f h_{gp}(x) \geq -\alpha(h_{gp}(x)) \), is satisfied. Given Assumption (3), the set \( S \) is nonempty, for any \( x \in \partial S \), \( h_{gp}(x) = 0 \) holds. As a result, \( \alpha(h_{gp}(x)) = 0 \) which gives \( L_f h_{gp}(x) \geq 0 \implies h_{gp} \geq 0 \). By applying Nagumo’s theorem [1], which states that for any \( C^1 \) function \( h_{gp} \), the condition \( h_{gp} \geq 0 \) on \( \partial S \) is necessary and sufficient for the set \( S \) to be forward invariant, completes the proof.

**Remark 3:** For the case when the kernel is only \( M \) times differentiable, and not infinitely MS differentiable, we require that \( M > 2\rho \), where \( \rho \in \mathbb{N} \) is the relative degree of the system. The proof above holds trivially for an \( M \) times differentiable kernel using Theorem 2.

We are interested in ensuring forward invariance of \( S \) characterized by \( h_{gp} \) for the system defined by (1). The admissible control space for the Gaussian CBF is given by,

\[
K_{gbf} = \{ u \in \mathbb{R}^m \mid L_f h_{gp}(x) + L_u h_{gp}(x) u + \alpha(h_{gp}(x)) \geq 0 \}. \tag{13}
\]

**Proposition 2:** Given a Gaussian CBF \( h_{gp}(x) : S \rightarrow \mathbb{R} \) defined by (11), where \( S \) is nonempty (8), any Lipschitz continuous controller \( u \in \mathbb{R}^m \), that satisfies (13) for any \( x \in \mathbb{R}^n \), renders \( S \) forward invariant for the system (1).

**Proof:** \( h_{gp} \) is a Gaussian process with an infinitely MS differentiable kernel. Using Theorem 2, \( h_{gp} \) is also infinitely MS differentiable and is, therefore, smooth. Since \( h_{gp} \) satisfies (13), we have \( L_f h_{gp}(x) + L_u h_{gp}(x) u \geq -\alpha(h_{gp}(x)) \). Using Theorem 1, the proof is complete.

### B. LIE DERIVATIVES OF GAUSSIAN CBF

The Gaussian CBF uses kernels for determining safety belief and uncertainty in the state space. As stated earlier, we use the SE kernel (4) which is an infinitely mean-square differentiable function. Computing its Lie derivatives is necessary for rectifying the control input and ensuring forward invariance for the system in the safe set. First, we take the partial derivative of (12) with respect to \( x \) at a query point \( x_q \),

\[
\frac{\partial h_{gp}(x)}{\partial x} \bigg|_{x_q} = \frac{\partial \mu(x)}{\partial x} \bigg|_{x_q} - \frac{\partial \sigma^2(x)}{\partial x} \bigg|_{x_q} = \mathbf{y}_N^\top \mathbf{K}^{-1} \frac{\partial \mathbf{k}(x)}{\partial x} \bigg|_{x_q} + 2\mathbf{k}(x)^\top \mathbf{K}^{-1} \frac{\partial \mathbf{k}(x)}{\partial x} \bigg|_{x_q}. \tag{14}
\]
The kernel derivative in (14) is given by,
\[
\frac{\partial k(x)}{\partial x} \bigg|_{x_q} = (x_{i1} - x_q)^\top k(x_{i1}, x_q)L^{-2},
\]
where \(k(x)\) is the \(i\)th element of \(k(x)\), and (15) is the \(i\)th row of \(\frac{\partial k(x)}{\partial x} \in \mathbb{R}^{N \times n}\). Now, we can compute the Lie derivatives of \(h\_{gp}(x)\) by taking its time derivative as follows,
\[
h\_{gp}(x) = \frac{\partial h\_{gp}(x)}{\partial x} f(x) + \frac{\partial h\_{gp}(x)}{\partial x} g(x)u = L_f h\_{gp}(x) + L_g h\_{gp}(x)u,
\]
where (14) is used in the Lie derivatives, \(L_f h\_{gp}(x)\) and \(L_g h\_{gp}(x)\). In prior literature, the derivative predictions of GP posterior mean and variance are exploited [36], [37]. Here, we take the partial derivatives of \(\mu\) and \(\sigma^2\) with respect to the state \(x\) which are different from the derivative predictions of GP posterior mean and variance.

### C. ONLINE SAFETY CONTROL

Consider a nominal control input \(u_{nom} \in \mathbb{R}^m\) that is designed as the feedback policy for system (1). This control policy may not restrict the solution of system (1) inside the safe set. An online quadratic program (QP) rectifies \(u_{nom}\) whose constraints are given by the Lie derivatives in (16) [2]. The QP optimization routine is set up as follows:

**Gaussian CBF-QP: Input modification**

\[
u_{rect} = \arg \min_{u \in \mathbb{R}^m} \frac{1}{2} \| u - u_{nom}\|^2 \text{ s.t. } L_f h_{gp}(x) + L_g h_{gp}(x)u + \alpha(h_{gp}(x)) \geq 0,
\]

where \(u_{rect}\) is the rectified control input. The QP constraint above ensures that the nominal control is followed as long as the safety condition is not violated, i.e., \(h_{gp}(x) \geq 0\). When approaching the boundary of the safe set, i.e., \(h_{gp} \to 0\), the QP rectifies \(u_{nom}\) minimally to \(u_{rect}\). By rectifying the control policy, the system is guaranteed to remain forward invariant for the safe set \(S\) due to Proposition 2. When solving for the QP, every term in the constraint is simply a numerical value except for the decision variable, the control input, which is rectified. Therefore, \(h_{gp}(x)\) being highly non-linear and non-convex does not affect finding the rectified control input. The algorithm for computing safe control input from the synthesized Gaussian CBF is shown in Algorithm 1.

**Remark 4:** Note that due to the non-parametric nature of the Gaussian CBF, the algorithm above can be treated as a blackbox routine. This is a beneficial property since, if a traditional CBF is altered, then the corresponding Lie derivatives also change explicitly in their form. However, in the Gaussian CBF, the structure of the Lie derivatives remains the same, i.e., the partial derivatives are explicitly agnostic to the underlying CBF. It is characterized only by the data and the dynamical system.

### D. GAUSSIAN CBF WITH NOISY QUERY STATE

Here, we extend Gaussian CBFs to handle the case when the query state, \(x_q\), is stochastic and therefore a random variable.

**Assumption 4:** The query state \(x_q\) is Gaussian distributed, \(x_q \sim \mathcal{N}(\mu, \Sigma)\), where \(\mu\) is the mean and \(\Sigma\) is its noise covariance matrix.

This has practical significance because accurate estimates of these states are required to generate safe control actions. In practice, however, measurement uncertainty is pervasive leading to error in the state estimates, thus degrading the safety behavior. As a result, we need to modify the posterior predictions of the GP in order to account for this noise.

The predictive equations for a Gaussian test input have been looked at before [38], [39]. Generally, if a Gaussian input is multiplied with the nonlinear GP predictive distribution, the resulting distribution is non-Gaussian,

\[
p(h_{gp}(x_q)|\mu, \Sigma) = \int p(h_{gp}(x_q)|x_q)p(x_q|\mu, \Sigma)dx_q.
\]

As a result, moment matching is used to derive the posterior predictions. To determine the moments of the predictive function value, both the query distribution and the distribution of the function given by the GP are averaged over. For the SE kernel, the posterior mean and variance can be computed for the predictive distribution in (18) in closed-form [40]. By using the law of iterated expectations, the posterior mean with a noisy query point \(x_q\) is given as follows [40],

\[
\mu(x_q) = q(x_q)^\top \mathbf{K}^{-1} y_N,
\]

where \(q = [q_1, \ldots, q_N]^\top \in \mathbb{R}^N\) with each \(q_i\) representing the expected covariance between \(h_{gp}(x_q)\) and \(h_{gp}(x_i)\),

\[
q_i(x_i, x_q) := \int k(x_i, x_q)N(x_q|\mu, \Sigma)dx_q = \sigma^2 |\Sigma|^{-1} \mathbf{I}_n + \mathbf{I}_n |\Sigma|^{-1} \exp \left( -\frac{1}{2}(x_i - \mu)^\top \mathbf{R}^{-1}(x_i - \mu) \right).
\]

This statement holds true for all kernels, in particular the SE, polynomial, and trigonometric kernels, if the integral of the kernel multiplied with a Gaussian distribution can be solved analytically.

---

**Algorithm 1** Gaussian CBF Synthesis & Safe Control

**Input:** GP Prior \(h_{gp}(x) \sim \mathcal{GP}(0, k(x, x'))\)

**System (1)**

Nominal Input \(u_{nom}\)

**Output:** Rectified input \(u_{rect}\)

1: **procedure** SafeControl
2: Sample \(X_N \leftarrow x_q \& y_N \leftarrow y\) using (10)
3: Synthesize \(h_{gp}(X_N, y_N)\) using (12)
4: Compute \(\frac{\partial h_{gp}(x)}{\partial x}\) using (14-15)
5: Setup QP constraint using (16)
6: Rectify \(u_{nom}\) using (17)
7: **return** \(u_{rect}\)
where \( \mathbf{R} = \Sigma + \mathbf{L}^2 \in \mathbb{R}^{n \times n} \). It is interesting to note the case for a deterministic query point \( \mathbf{x}_q \), where \( \Sigma \mathbf{D} = 0 \) and \( \mu = \mathbf{x}_q \). On comparing (19) with (5), the posterior mean for the noisy input results in the same posterior mean for the noise-free input, since \( q_i(\mathbf{x}_q, \mathbf{x}_q) \) collapses to \( k_i(\mathbf{x}_q, \mathbf{x}_q) \) in (4). Effectively, the noise-free input point is a special case of the noisy posterior prediction with \( \Sigma = 0 \) and \( \mu = \mathbf{x}_q \).

For details on the derivation of the predictive variance for the noisy test point, see [40]. Here, we simply state the posterior predictive variance which is as follows,

\[
\sigma^2(\mathbf{x}_q) = \sigma^2_f - \text{tr}(\mathbf{K}^{-1} \mathbf{V}) + \beta^T (\mathbf{V} \beta - \mathbf{q}).
\] (20)

with \( \beta = \mathbf{K}^{-1} \mathbf{y}_N \in \mathbb{R}^N \), and the entries of \( \mathbf{V} \in \mathbb{R}^{N \times N} \) are given by,

\[
v_{ij} = \frac{k(x_i, \mu) k(x_j, \mu)}{2\Sigma L^2 + \mathbf{I}_n} \exp((z_{ij} - \mu)^T \mathbf{T}(z_{ij} - \mu)),
\]

where \( \mathbf{T} := (\Sigma + \frac{1}{2} \mathbf{L}^2)^{-1} \Sigma \mathbf{L}^{-2} \in \mathbb{R}^{n \times n} \) and each entry in \( z_{ij} := \frac{1}{2}(x_i + x_j) \in \mathbb{R}^n \). As seen in (19) and (20), both the predictive mean and variance explicitly depend on the mean \( \mu \) and the covariance matrix \( \Sigma \) of the Gaussian distributed query state \( \mathbf{x}_q \).

The Gaussian CBF for a noisy query point \( \mathbf{x}_q \) is given by,

\[
h_{\text{BP}}(\mathbf{x}_q) := \mu(\mathbf{x}_q) - \sigma^2(\mathbf{x}_q).
\] (21)

To compute the Lie derivatives, the partial derivative of (21) with respect to \( \mu \) is given by,

\[
\frac{\partial \mu(\mathbf{x}_q)}{\partial \mathbf{x}_q} \bigg|_\mu = \frac{y_N}{\mathbf{K}^{-1}} \frac{\partial q(x)}{\partial \mathbf{x}} \bigg|_\mu
\]

\[
\left[ \frac{\partial \sigma^2(\mathbf{x}_q)}{\partial \mathbf{x}_q} \right]_{\mu_k} = \sum_{i=1}^{N} \left( k_{ij} \frac{\partial v_{ij}}{\partial x_k} \right)_{\mu_k} - \sum_{i=1}^{N} k_{ij} \frac{\partial q_i}{\partial x_k} \bigg|_{\mu_k},
\] (23)

where \( k = \{1, \ldots, n\} \), \( x_k \) and \( \mu_k \) are the \( k \)th entries of \( \mathbf{x}_q \) and \( \mu \), respectively, and \( k_{ij} = k(x_i, x_j) \in \mathbb{R} \). We emphasize that \( \mathbf{V} \) is a symmetric matrix which allows combining the summation with the choice of indices in (23). The Lie derivatives can then be computed using the equations above, similar to (16), to achieve safe constrained control by setting up a QP as constructed in (30).

V. APPLICATION TEST CASE: QUADROTOR

To demonstrate the efficacy of our method, we implement our proposed technique on a quadrotor system. Quadrotors pose an interesting and challenging problem due to their inherently unstable nature. We run 3 separate experiments using Gaussian CBF on a quadrotor: (a) safe constrained control for arbitrary safe sets (b) explore the state space safely and synthesize the safe set online, and (c) safe constrained control in presence of noisy states and compare the performance with regular CBFs. First, we review quadrotor dynamics followed by safety rectification using the Gaussian CBF for a quadrotor platform.

A. QUADROTOR DYNAMICS

We consider the position dynamics and attitude dynamics of a quadrotor model evolving in a coordinate-free framework. This framework uses a geometric representation for its attitude given by a rotation matrix \( \mathbf{R} \) on \( SO(3) := \{ \mathbf{R} \in \mathbb{R}^{3 \times 3} | \mathbf{R}^T \mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = 1 \} \). \( \mathbf{R} \) represents the rotation from the body-frame to the inertial-frame. The origin of the body-frame is given by the quadrotor’s center of mass, denoted by \( \mathbf{r} \in \mathbb{R}^3 \). A quadrotor is an underactuated system since it has 6 DOF, due to its configuration space being \( SE(3) := \mathbb{R}^3 \times SO(3) \), but 4 control inputs; thrust \( F \in \mathbb{R} \) and moments \( \mathbf{M} \in \mathbb{R}^3 \). The equations of motion are:

\[
\dot{\mathbf{r}} = \mathbf{v},
\]

\[
m \dot{\mathbf{v}} = -mg \mathbf{e}_3 + F \mathbf{R} \mathbf{e}_3,
\] (24)

\[
\dot{\mathbf{R}} = \mathbf{R} \Omega \times,
\]

\[
\dot{\mathbf{J}} \dot{\Omega} = \mathbf{M} - (\Omega \times \mathbf{J} \dot{\Omega}),
\] (25)

where \( \mathbf{v} \in \mathbb{R}^3 \) is the velocity in the inertial frame, \( m \in \mathbb{R} \) is the quadrotor mass, \( g \in \mathbb{R} \) is gravity, \( \mathbf{e}_3 = [0 0 1]^T \in \mathbb{R}^3 \), \( \Omega \in \mathbb{R}^3 \) is the body-frame angular velocity, \( \mathbf{J} \in \mathbb{R}^{3 \times 3} \) is the inertia matrix, and \( \cdot \times : \mathbb{R}^3 \rightarrow so(3) \) is the skew-symmetric operator, such that \( \mathbf{v} \times \mathbf{y} = \mathbf{x} \times \mathbf{y} \).

B. SETPOINT GENERATION FOR QUADROTOR

For achieving safety constrained control of the quadrotor, we first compute setpoints. These setpoints are sent to a Crazyflie 2.1 [41], in the form of desired thrust, \( F_{\text{des}} \), and desired roll, pitch, yaw angles, \( \eta = [\phi_{\text{des}}, \theta_{\text{des}}, \psi_{\text{des}}]^T \in \mathbb{R}^3 \). The Crazyflie is equipped with a fast response low-level onboard controller that can directly track these setpoint commands. More details regarding the hardware experimental setup are covered in Section VI-A.

Given a desired trajectory, \( \mathbf{r}_{\text{des}} \in \mathbb{R}^3 \), that is twice differentiable, a second-order integrator model is set up,

\[
\begin{bmatrix}
\dot{\mathbf{r}} \\
\dot{\mathbf{x}}
\end{bmatrix} =
\begin{bmatrix}
0 & I \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{r} \\
\mathbf{x}
\end{bmatrix} +
\begin{bmatrix}
0 \\
0
\end{bmatrix}
\mathbf{u},
\] (26)

where \( \mathbf{x} = [\mathbf{r} \mathbf{F}]^T \in \mathbb{R}^6 \) and \( \mathbf{u} = \mathbf{r}_{\text{des}} \in \mathbb{R}^3 \). The input \( \mathbf{u} \) in (26) is rectified using the synthesized Gaussian CBF generating the following rectified setpoints,

\[
\phi_{\text{rect}} = \frac{(u_1 \text{rect} \sin \psi - u_2 \text{rect} \cos \psi)}{g},
\] (27)

\[
\theta_{\text{rect}} = \frac{(u_1 \text{rect} \cos \psi + u_2 \text{rect} \sin \psi)}{g},
\] (28)

\[
F_{\text{rect}} = m(u_3 \text{rect} + g),
\] (29)

where the desired yaw is assumed to be zero and small angle approximations are made to invert the dynamics in (24) for simplicity [42], [43]. Next, we discuss the safety rectification of \( \mathbf{u} \) to compute \( \mathbf{u}_{\text{rect}} \) using the Gaussian CBF.

C. ONLINE CONTROL RECTIFICATION

Given a Gaussian CBF expressed in the position space, the relative degree for system (26) is \( \rho = 2 \). The associated Lie
derivatives for the Gaussian CBF in (12) are,
\[
L_f h_{gp}(x) = \left( \nabla \mu(x) - \nabla \sigma^2(x) \right)^T f(x),
\]
\[
L_f^2 h_{gp}(x) = f(x)^T \left( \mathbf{H}_{\mu}(x) - \mathbf{H}_{\sigma^2} \right) f(x)
\]
\[
+ \left( \nabla \mu(x) - \nabla \sigma^2(x) \right)^T \nabla f(x) \cdot f(x),
\]
\[
L_g L_f h_{gp}(x) = f(x)^T \left( \mathbf{H}_{\mu}(x) - \mathbf{H}_{\sigma^2}(x) \right) g(x)
\]
\[
+ \left( \nabla \mu(x) - \nabla \sigma^2(x) \right)^T \nabla f(x) \cdot g(x),
\]
where \( \nabla \mu(x) = \frac{\partial \mu(x)}{\partial x} \) and \( \nabla \sigma^2(x) = \frac{\partial \sigma^2(x)}{\partial x} \) are the gradients of GP mean and variance in (14) and \( \nabla f(x) = \frac{\partial f(i)}{\partial x} \) is the Jacobian of \( f(x) \). \( \mathbf{H}_{\mu}(x) \) and \( \mathbf{H}_{\sigma^2}(x) \) are the Hessians of GP mean and variance given by,
\[
\mathbf{H}_{\mu}(x) = \left( \sum N_i a_i \frac{\partial^2 k_i(x)}{\partial x^2} \right),
\]
\[
\mathbf{H}_{\sigma^2}(x) = -2 \nabla k(x) \mathbf{K}^{-1} \nabla k(x) - 2 \left( \sum N_i b_i \frac{\partial^2 k_i(x)}{\partial x^2} \right),
\]
where \( a_i \) is the \( i \)th entry of \( \nabla k(x) \mathbf{K}^{-1} \in \mathbb{R}^{1 \times N} \), \( b_i \) is the \( i \)th entry of \( k(x) \mathbf{K}^{-1} \in \mathbb{R}^{1 \times N} \), \( \nabla k(x) = \frac{\partial k_i(x)}{\partial x} \in \mathbb{R}^{N \times N} \), and \( \frac{\partial^2 k_i(x)}{\partial x^2} \) is the partial derivative of (15) with respect to \( x \). For the case when the query state is noisy, \( \mathbf{x}_{\text{nom}} \), we use the predictive mean and variance (21) as described in Section IV-D. Similarly, the corresponding partial derivatives are used to compute the Jacobians and Hessians from (22)-(23) to compute the Lie derivatives for the noisy query state. Given the nominal control input \( \mathbf{u}_{\text{nom}} \in \mathbb{R}^3 \) in (26), the QP below rectifies \( \mathbf{u}_{\text{nom}} \) into \( \mathbf{u}_{\text{rect}} \in \mathbb{R}^3 \).

**Gaussian CBF-QP: Input modification**
\[
\mathbf{u}_{\text{rect}} = \arg \min_{\mathbf{u} \in \mathbb{R}^3} \frac{1}{2} \| \mathbf{u} - \mathbf{u}_{\text{nom}} \|^2 \; \text{s.t.} \; \quad L_f^2 h_{gp}(x) + L_g L_f h_{gp}(x) \mathbf{u} + \mathcal{K}^T \mathcal{H} \geq 0, \quad (30)
\]

where \( \mathcal{K} = [k_1 \ k_2]^T \in \mathbb{R}^2 \) is the coefficient gain vector, and \( \mathcal{H} = [L_f h_{gp}(x) \ h_{gp}(x)]^T \in \mathbb{R}^2 \) is the Gaussian Lie derivative vector. The rectified input \( \mathbf{u}_{\text{rect}} \) is used to compute the rectified setpoints using (27), (28), (29) which are then ultimately sent to the quadrotor.

**VI. EXPERIMENTAL VERIFICATION**

In this section, we discuss the implementation of our method on a hardware quadrotor. We test our proposed formulation in three different scenarios. In the first setting, we demonstrate safe constrained control, where the Gaussian CBF is used to formulate the candidate function. These safe sets are arbitrarily designed and are not limited to taking any convex shape. For the second demonstration, we synthesize the safety function online by exploring the state space while avoiding static collisions. The quadrotor performs safe control within the constructed Gaussian CBF. And for the final scenario, we revisit the constrained control problem for a given candidate function, but in the presence of noisy position states. We compare the safe controlled behavior with a regular CBF. All experiments can be seen here: https://youtu.be/HX6uokvCiGk.

**A. EXPERIMENT SETUP**

We use the Crazyflie 2.1 as the hardware quadrotor. State estimation is performed onboard with the help of an external low-cost lighthouse positioning system [41]. All computations are done remotely on a ground station equipped with an Intel i7-9800X at 4.4 GHz processor and 16 GB RAM. The crazyflie_ros API is used to communicate for interprocess communication, subscribing to pose information, and publishing setpoints over the Crazyradio PA USB dongle [44]. Positions and velocities are collected at 20 Hz with a data capacity set to 300 samples. Gaussian CBF synthesis and rectification routine (30) are run on a parallel thread at 50 Hz where solving the QP takes under 5ms. Nominal setpoint commands are sent to the Crazyflie at 100 Hz with the help of a Logitech joystick controller, which acts as the nominal controller in the QP formulation.

**B. SCENARIO A: SAFE CONTROL FOR ARBITRARY SAFE SETS**

The objective in this scenario is to demonstrate safe constrained control for any given arbitrary safe set using the Gaussian CBF formulation. We assume a high-level observer or planner provides a map from which we can sample (un)safe locations. For instance, take the example of a satellite view for a street or the indoor map of a warehouse unit, where the goal is to navigate an autonomous agent safely and provide safety specifications at the planning phase. The configuration space for position safety in such settings cannot be designed by hand effectively. By using the data driven design of Gaussian CBFs, we can construct safe sets based on the dataset allowing flexible realizations of safe sets based on information from a high-level planner or observer.

In this scenario, we construct a safety map in 2D, where the domain is chosen to be \([-0.35, 0.35]\) along each lateral axis, \( x \) and \( y \). We uniformly sample, \( N = 200 \), \((x, y)\) input points. The safety sample for each input coordinate is drawn from a uniform distribution, \( y \sim \mathcal{U}(a, b) \), where \( a = -1.0 \) and \( b = 2.0 \) are the lowest and highest values respectively of the distribution. This gives a discrete 2D safety map, where for each of the 200 \((x, y)\) coordinates, we have an associated target safety sample. The safety maps are synthesized once and do not change during the experiment, so the sampling distance is set to \( \tau = 0 \). The hyperparameters are arbitrarily chosen to generate arbitrary safety maps: \( \mathbf{L} = \text{diag}(0.1, 0.1), \sigma_f = 1, \sigma_r = 0.01 \). The Gaussian CBF characterizes the posterior safety map as follows,
\[
h_{gp}(x) := \mu(x) - 4\sigma^2(x). \quad (31)
\]
We generate 3 arbitrary safe sets and run 3 separate experiments with the quadrotor always starting in the safe set as shown in Figure 3. For each experiment, we plot the flight trajectory of the quadrotor, its initial and final positions, and the 0-level set of $h_{gp}$. The plot of $h_{gp}(t)$ is also shown for each experimental run. First, we point out that the data generates arbitrary non-convex safe sets. The posterior mean represents the safety belief in $h_{gp}$, whereas the posterior variance quantifies the notion of safety for regions in the state space where we have few or no samples. We can thus generate very safe realizations of candidate CBFs with high probabilistic bounds. The flight trajectory of the quadrotor always remains inside the safe set based on the QP formulation in (30). This can be verified by looking at the $h_{gp}(t)$ for each experiment, which is always non-negative.

C. SCENARIO B : ONLINE SYNTHESIS OF SAFE SET WITH OBSTACLE AVOIDANCE

The objective is to synthesize the safety function online by exploring the state space and avoiding collisions. This has great practical significance in safe navigation since onboard sensors are limited in collecting data only within their local proximity. Therefore, we cannot know a priori the complete safety map. Moreover, the sensed data will also need to alter the safety decision boundary online. With the help of Gaussian CBFs, we can incrementally change the safe set as more data is collected. This allows expansion of the safe set in a non-convex manner, which is required in many practical scenarios involving unstructured environments.

In this scenario, $h_{gp}(x)$ denotes the distance between the quadrotor and the obstacle. Here, we use two obstacles and therefore have two separate distance measurements,

$$
d_a = (r_{xy} - p_a)^\top (r_{xy} - p_a) - R_a^2,
$$

$$
d_b = (r_{xy} - p_b)^\top (r_{xy} - p_b) - R_b^2,
$$

where $r_{xy}$ is the quadrotor’s lateral position, similarly $p(\cdot) \in \mathbb{R}^2$ is the obstacle’s lateral position and $R(\cdot) \in \mathbb{R}$ is the obstacle radius. The overall safety sample is taken as the noisy estimate by combining the two distance measurements,

$$
y := d_a \cdot d_b + w, \quad w \sim N(0, \sigma_y^2),
$$

where $\sigma_y^2 \in \mathbb{R}$ is the noise variance. We take noisy sample observations to make the experiment more realistic. Moreover, we also wanted to highlight experimentally that, despite using noisy safety samples, our approach is robust enough to design non-convex safe sets online and ensure the system remains safe. If the quadrotor is closer to one of the obstacles, the product decreases, as a result reducing the safety metric. Note that, even though the obstacles are assumed to be convex, the final safe set constructed need not be convex. This is due to the noisy distance measurements observed and the posteriors being constructed online. The sampling distance is set to $\tau = 0.1$ and the hyperparameters are optimized.

**FIGURE 3.** Experiments are run for arbitrary safe sets generated using the Gaussian CBF $h_{gp}$. The initial (■), mid-flight (●), and final (•) quadrotor positions are shown. The 0-level contour line is marked (bold gray). The temporal plot of $h_{gp}$ for each experiment shows that the quadrotor always remains inside the safe set.
by maximizing the log marginal likelihood using gradient methods [32]. We use the same Gaussian CBF as (31) in scenario A to generate the posterior safe set online.

The quadrotor starts in an initial safe set containing only the initial position, see Figure 4. The quadrotor collects safety samples $y$ with the corresponding state $x$ along its trajectory. With the data being collected, the safety function $h_{gp}(x)$ and its associated safe set is constructed online using (31). In regions where we have data, the safety belief is high and safety uncertainty is low. As more data is collected, the associated safe set expands. For unexplored regions in the state space, the safety uncertainty is high due to the high posterior variance. This aligns with the intuition that safety is not known with high confidence in unexplored spaces.

As seen in Figure 4, we see two sets of data samples during the quadrotor’s flight. Initially, the quadrotor explored a small region in the state space containing 15 samples. The 0-level set is shown in bold white. Then the quadrotor continues exploring the state space further, thus expanding the safe set. Notice that the expansion of the safe set is not limited to any convex expansion. The 0-level set for the larger safe set of all 31 samples collected thus far is marked with dashed black line in Figure 4. The exploration process continues, and the quadrotor is able to detect regions in the state space which are unsafe, particularly, when it gets closer to the obstacles. The final safety map for the Gaussian CBF using (12) is shown in Figure 6.

**D. SCENARIO C : SAFE CONTROL IN PRESENCE OF NOISY STATE**

For the final experiment, we consider the problem of safe constrained control in the presence of noisy position states.

In many practical applications, measurement noise is a common occurrence which can degrade system performance and lead to unsafe consequences. This is a particularly hard problem because we consider noise for both the input and observations to the GPs. In our current scenario, this would be noise for the safety samples and the system query state, in particular, the position state of the quadrotor. The query state is given by $x_q := \begin{bmatrix} r^T & \dot{r}^T \end{bmatrix}^T$, where $r \sim \mathcal{N}(r, \Sigma)$ is the Gaussian distributed noisy position state. We have observability of the noisy query state $x_q$ and assume knowledge of the noise covariance matrix $\Sigma$ for the position states.

The safety objective is to keep the quadrotor inside a circle of radius $D$. We first construct a standard CBF using the following candidate function,

$$h_{cbf} := D^2 - x^2 - y^2,$$
where \( D = 0.35 \) is the safety boundary radius. Next, 100 samples are sampled randomly with Gaussian noise from this candidate CBF, \( y = \mathcal{N}(h_{cbf}, 0.03) \). The posterior GP is then computed, along with hyperparameter optimization, using (31) from this dataset which forms the Gaussian CBF. Both the CBFs as well as safety samples are shown in Figure 7. In this scenario we choose \( \tau = 0 \), since the safe sets are fixed and require no online sampling.

We perform 3 separate experiments on the quadrotor for the standard CBF using different values for the noisy position states. The different noise covariance values in the experiments are \( \Sigma = [0.015, 0.025, 0.04] I_3 \in \mathbb{R}^{3 \times 3} \). For each experiment in Figure 8, the quadrotor starts inside the safe set and then violates safety at the boundary when subjected to noisy position states. As the value of the noise increases, the quadrotor exhibits more violation of the safety constraint by going outside the safety boundary radius. Since CBFs rectify the control input pointwise and do not account for any measurement noise in its formulation, measurement noise in the position states degrades the safety performance. For each experiment, the temporal behavior of \( h_{cbf}(t) \) verifies that the safety constraint is violated due to the negative values.

We next look at the experiments using Gaussian CBFs for the same values of noise covariance used above. In Figure 9, we see that for every experiment, the quadrotor remains confined within a more conservative safe set, which is inside the primary safety boundary radius. This occurs because in the presence of noisy input (query) to the GPs, the posterior mean uses a more conservative weighted kernel \( q \) in (21), whose entries \( q_i \) have coefficients lesser than the coefficients of the SE kernel.
FIGURE 9. The quadrotor does not go outside the circular boundary of $R = 0.35$ for Gaussian CBF because a more conservative safe set is modeled by $h_{gp}$. When looking at $h_{gp}(t)$ and $h_{cbf}(t)$ on the ground truth position, the quadrotor is further away from the boundary of the safe set using $h_{cbf}$. The coefficient of an entry $q_i$ given by $\sigma_i^2 \Sigma L^{-2} + I_n \frac{1}{2}$ is always lesser than $\sigma_i^2$ since the eigenvalues of $\Sigma L^{-2} + I_n$ are always greater than 1 ($\Sigma L^{-2}$ is a positive definite matrix added to the identity matrix). Therefore, the determinant is always positive and greater than unity. Intuitively, this makes sense since the GP posterior distribution is not overfitting to the noisy input query states, thus leading to a more conservative posterior estimation. We also plot $h_{cbf}$ as a function of time using the trajectory of the quadrotor rectified under $h_{gp}$. The quadrotor does not get close to the circular boundary, since it is constrained conservatively by $h_{gp}$, thus ensuring that the original safety requirement is met. Indeed, if the noise becomes very large, then the safe set may not exist under $h_{gp}$. Determining the bounds on the measurement noise is currently outside the scope of this study and is left for future investigation. Here, we are primarily interested in achieving safe control in the presence of noisy query states with nonempty compact safe sets.

E. NOTE ON COMPLEXITY

GPs are known to scale cubically with data i.e., $O(N^3)$, where $N$ is the number of datapoints. This complexity arises due to the inverse operation in (12) for the covariance matrix $K$. As the number of data points increases, this could potentially cause a computational bottleneck. We address this with the help of rank-1 inverse method. For example, given 500 samples, it only takes 25ms to synthesize the Gaussian CBF. Thereafter, we handle more datapoints by performing rank-1 inverse approximations giving tremendous boost in computational speed. For every new data point included, it only takes 4ms to compute the inverse covariance matrix.

VII. CONCLUSION

In this study, we proposed a framework for the synthesis of a safety function in a data-driven manner using GPs. The formulation requires safety samples as opposed to the traditional requirement of a smooth function. The newly formulated CBF called the Gaussian CBF was constructed by using a flexible GP prior. GPs provide the posterior mean and variance which serve as analogues for safety belief and uncertainty in our methodology. By exploiting the kernel properties in the posterior mean and variance, we were able to analytically compute the associated Lie derivatives. The Lie derivatives served as constraints in formulating a QP for rectifying the given nominal control input. We empirically verified our framework on a hardware quadrotor platform without risking any expensive system failures. We verify our approach on three different scenarios. The objective in each experiment was to synthesize a candidate safety function using GPs. We successfully show safe control for arbitrary safe sets synthesized using Gaussian CBFs, online synthesis of a Gaussian CBF as more data is collected in a collision avoidance problem, and juxtapose a Gaussian CBF with a regular CBF for constrained control in the presence of noisy position states. The quadrotor always remained inside the safe sets associated with the synthesized Gaussian CBFs. As part of future work, we would like to delve deeper and develop
Gaussian CBFs for stochastic settings as well, and compare against other CBPF techniques.

ACKNOWLEDGMENT

The authors would like to acknowledge Vishal Murali and Akash Patel for their invaluable assistance and discussions.

REFERENCES

[1] F. Blanchini and S. Miani, Set-Theoretic Methods in Control, Springer, 2008.
[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” IEEE Trans. Autom. Control, vol. 62, no. 8, pp. 3861–3876, Aug. 2016.
[3] 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. Autom. (ICRA), May 2017, pp. 3293–3298.
[4] S.-C. Hsu, X. Xu, and A. D. Ames, “Control barrier function based quadratic programs with application to bipedal robotic walking,” in Proc. Amer. Control Conf. (ACC), Jul. 2015, pp. 4542–4548.
[5] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A Framework for worst-case and stochastic safety verification using barrier certificates,” IEEE Trans. Autom. Control, vol. 52, no. 8, pp. 1415–1428, Aug. 2007.
[6] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” IFAC Proc. Volumes, vol. 40, no. 12, pp. 462–467, 2007.
[7] U. Borrman, L. Wang, A. D. Ames, and M. Egerstedt, “Control barrier certificates for safe swim behavior,” IFAC-PapersOnLine, vol. 48, no. 27, pp. 68–73, 2015.
[8] T. Ibuik, S. Wilson, A. D. Ames, and M. Egerstedt, “Distributed collision-free motion coordination on a sphere: A conic control barrier function approach,” IEEE Control Syst. Lett., vol. 4, no. 4, pp. 976–981, Oct. 2020.
[9] G. Wu and K. Sreenath, “Safety-critical control of a planar quadrotor,” in Proc. Amer. Control Conf. (ACC), Jul. 2016, pp. 2252–2258.
[10] G. Wu and K. Sreenath, “Safety-critical control of a 3D quadrotor with range-limited sensing,” in Proc. Dyn. Syst. Control Conf., 2016, Art. no. V001T05A006.
[11] F. Berkenkamp, R. Moriconi, A. Schoellig, and A. Krause, “Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes,” in Proc. IEEE 55th Conf. Decis. Control (CDC), Dec. 2016, pp. 4661–4666.
[12] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans, “Automatic gait optimization with Gaussian process regression,” in Proc. IJCAI, vol. 7, 2007, pp. 944–949.
[13] M. Tesch, J. Schneider, and H. Choset, “Using response surfaces and expected improvement to optimize snake robot gait parameters,” in Proc. IEEERSJ Int. Conf. Intell. Robots Syst., Sep. 2011, pp. 1069–1074.
[14] F. Berkenkamp, A. P. Schoellig, and A. Krause, “Safe controller optimization for quadrotors with Gaussian processes,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2016, pp. 491–496.
[15] A. A. Ahmadi and A. Majumdar, “Some applications of polynomial optimization in operations research and real-time decision making,” Optim. Lett., vol. 10, no. 4, pp. 709–729, Apr. 2016.
[16] A. A. Ahmadi and A. Majumdar, “DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization,” SIAM J. Appl. Algebra Geometry, vol. 3, no. 2, pp. 193–230, Apr. 2019.
[17] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, “Synthesis of control barrier functions using a supervised machine learning approach,” in Proc. IEEERSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2020, pp. 7139–7145.
[18] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, “Learning control barrier functions from expert demonstrations,” in Proc. 59th IEEE Conf. Decis. Control (CDC), Dec. 2020, pp. 3717–3724.
[19] A. Robey, L. Lindemann, S. Tu, and N. Matni, “Learning robust hybrid control barrier functions for uncertain systems,” IFAC-PapersOnLine, vol. 54, no. 5, pp. 1–6, 2021.
MOUHYEMEN A. KHAN received the B.S. degree in electrical and computer engineering from Texas A&M University, College Station, USA, in 2014. Currently, he is pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA, with a focus on dynamical system safety using control theoretic ideas and machine learning.

He co-founded the Qatar Robotics Institute for Development, in 2013 aimed for STEM development and robotics education in Qatar and served as its CEO, from 2013 to 2018. From November 2014 to May 2016, he was a Research Assistant with Qatar University and Carnegie Mellon University, Qatar, where he developed an indoor testbed for quadrotors to deploy and test coverage algorithms.

TATSUYA IBUKI (Member, IEEE) received the B.Eng., M.Eng., and Ph.D.Eng. degrees in control engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2008, 2010, and 2013, respectively.

He served as a Research Fellow at the Japan Society for the Promotion of Science, from 2012 to 2013. From 2013 to 2020, he was an Assistant Professor at the Department of Systems and Control Engineering, Tokyo Institute of Technology. Since 2020, he has been a Senior Assistant Professor with the Department of Electronics and Bioinformatics, Meiji University, Kanagawa. His research interests include cooperative control of robotic networks, fusion of control theory and machine learning, and vision-based estimation and control.

ABHIJIT CHATTERJEE (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1990.

He is a Professor with the School of Electrical and Computer Engineering, Georgia Tech. He has authored over 450 papers in refereed journals and meetings and has 22 patents. He is a co-founder of Ardext Technologies Inc.—a mixed-signal test solutions company. His research interests include error-resilient signal processing and control systems, mixed-signal/RF/multi-GHz design and test, and adaptive real-time systems. He was the Chairman and the Chief Scientist from 2000 to 2002. He received the NSF Research Initiation Award in 1993 and the NSF CAREER Award in 1995. He has received seven Best Paper Awards and three Best Paper Award nominations. His work on self-healing chips was featured as one of General Electric’s key technical achievements in 1992 and was cited by The Wall Street Journal. In 1995, he was named a Collaborating Partner in NASA’s New Millennium Project. In 1996, he received the Outstanding Faculty for Technology Transfer Award, also given by the Packaging Research Center. In 2007, his group received the Margarida Jacome Award for work on VIZOR: Virtually Zero Margin Adaptive RF from the Berkeley Gigascale Research Center (GSRC).

***