Pointwise Feasibility of Gaussian Process-based Safety-Critical Control under Model Uncertainty

Fernando Castañeda*, Jason J. Choi*, Bike Zhang, Claire J. Tomlin, and Koushil Sreenath

Abstract— Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs) are popular tools for enforcing safety and stability of a controlled system, respectively. They are commonly utilized to build constraints that can be incorporated in a min-norm quadratic program (CBF-CLF-QP) which solves for a safety-critical control input. However, since these constraints rely on a model of the system, when this model is inaccurate the guarantees of safety and stability can be easily lost. In this paper, we present a Gaussian Process (GP)-based approach to tackle the problem of model uncertainty in safety-critical controllers that use CBFs and CLFs. The considered model uncertainty is affected by both state and control input. We derive probabilistic bounds on the effects that such model uncertainty has on the dynamics of the CBF and CLF. Then, we use these bounds to build safety and stability chance constraints that can be incorporated in a min-norm convex optimization program, called GP-CBF-CLF-SOCP. As the main theoretical result of the paper, we present necessary and sufficient conditions for pointwise feasibility of the proposed optimization problem. We believe that these conditions could serve as a starting point towards understanding what are the minimal requirements on the distribution of data collected from the real system in order to guarantee safety. Finally, we validate the proposed framework with numerical simulations of an adaptive cruise controller for an automotive system.

I. Introduction

A. Motivation

Guaranteeing stability and safety is of major importance for the deployment of autonomous systems in the real world. Theory for model-based controllers for this purpose is well established. However, the models that these controllers rely on are usually inaccurate. Real systems can be hard to model, might have parameters that can be challenging to identify, or both. In the presence of such model uncertainty, the theoretical guarantees of stability and safety that model-based controllers provide can be lost. On the other hand, data-driven control techniques have the potential to solve difficult control tasks and are becoming increasingly popular. However, guarantees of safety or stability for these approaches typically require rich enough data to fully characterize the dynamics of the system [1], [2], [3]. In practice, collecting this data might not be feasible for systems with uncertain dynamics and uncertain input effects. For instance, it might require applying unstable control inputs that would

lead to unsafe behavior and possible damage of the system. Providing safety guarantees for such uncertain systems while using only data that is feasible to collect remains an open problem [4].

This paper takes important steps towards solving this problem by combining a model-based method with a data-driven approach. We use Control Barrier Functions (CBFs, [5]), a popular tool to enforce safety constraints [6]. However, CBF-based controllers suffer from the problem of model uncertainty, which we tackle using Gaussian Process (GP) regression. We build on our previous work [7], where we used GP regression to address model uncertainty in Control Lyapunov Functions (CLFs, [8]). We propose an optimization-based safety-critical controller, and we conduct an analysis of its feasibility. Results from this feasibility analysis show great promise for the design of online safe learning algorithms for uncertain systems.

B. Related Work

Model uncertainty of CBF-based controllers was first explored by the use of adaptive [9], [10] and robust [11], [12] control techniques. Recent work has demonstrated that CBF-based controllers combined with data-driven methods can be useful for coping with model uncertainty. With neural networks it is often difficult to obtain rigorous guarantees for reliable safety controllers, although several works show practical outcomes [13], [14], [15]. GP regression is an alternative approach that provides a probabilistic guarantee of its prediction. CLF or CBF-based controllers that use GP regression to learn the model uncertainty terms have been proposed before [16], [17], [18]. However, these works do not consider uncertainty in the input effects, which is significant for many systems [7]. In [19], this problem is tackled by using Matrix-Variate GP regression.

C. Contributions and Organization

First, we extend the GP-CLF-SOCP controller presented in [7] to the safety-critical control case by including an uncertainty-aware CBF chance constraint. This results in the formulation of the so-called *Gaussian Process-based Control Barrier Function and Control Lyapunov Function Second-Order Cone Program (GP-CBF-CLF-SOCP)*.

Secondly, the main theoretical result of this paper is a set of necessary, sufficient, and necessary and sufficient conditions for pointwise feasibility—feasibility at each timestep—of the proposed GP-CBF-CLF-SOCP controller. We believe that this analysis can serve as the first step towards the construction of specific conditions on the data distribution

collected from the real system to guarantee safety in systems with uncertain input effects and dynamics.

The rest of the paper is organized as follows. Sec. II briefly revisits the necessary background on CLFs and CBFs, and presents how model uncertainty affects their respective dynamics. In Sec. III we give an overview of GP regression. In Sec. IV we present the proposed probabilistic safe stabilizing optimization-based controller. In Sec. V we develop the necessary and sufficient conditions for feasibility of the proposed optimization problem. In Sec. VI we show the performance of the controller on numerical simulations of an adaptive cruise control system under model uncertainty. Finally, in Sec. VII we give concluding remarks.

II. BACKGROUND

In this section, we present the necessary background on CLFs and CBFs.

Consider a nonlinear control-affine system of the form:
$$\dot{x} = f(x) + g(x)u, \tag{1}$$
where $x \in \mathcal{X} \subset \mathbb{R}^n$ is the system state and $u \in \mathbb{R}^m$ is the control input. The vector fields f and g are assumed to be locally Lipschitz continuous, and without loss of generality we assume that the origin is the equilibrium point, $f(0) = 0$, that we want the system state to converge to. We will refer to system (1) as the *true plant*.

A. Control Lyapunov Functions

Definition 1. Let $V \colon \mathcal{X} \to \mathbb{R}_+$ be a positive definite, continuously differentiable, and radially unbounded function. V is a *Control Lyapunov Function* (CLF) for system (1) if for each $x \in \mathcal{X} \setminus \{0\}$ the following holds:
$$\inf_{u \in \mathbb{R}^m} \underbrace{L_f V(x) + L_g V(x)u}_{=\dot{V}(x,u)} < 0, \tag{2}$$
where $L_f V(x) := \nabla V(x) \cdot f(x)$ and $L_g V(x) := \nabla V(x) \cdot g(x)$ are Lie derivatives of V with respect to f and g, respectively.

If system (1) admits such a CLF, it is globally asymptotically stabilizable to the origin [8]. CLFs can also impose a stronger notion of stabilizability, which is the exponential convergence to the origin. If there exists a compact subset $D \subseteq \mathcal{X}$ that includes the origin such that for all $x \in D$ and for some constant $\lambda > 0$, it holds that
$$\inf_{u \in \mathbb{R}^m} L_f V(x) + L_g V(x)u + \lambda V(x) \leq 0, \tag{3}$$
and $\exists\, c_{exp} > 0$ such that $\Omega_{c_{exp}} := \{x \in D \subseteq \mathcal{X} : V(x) \leq c_{exp}\}$ is a sublevel set of $V(x)$, then the origin is locally exponentially stabilizable from $\Omega_{c_{exp}}$. In this case, we say that V is a *locally exponentially stabilizing* CLF. Condition (3) can be used as a constraint in a min-norm quadratic program (QP) [20]:

CLF-QP:
$$u^*(x) = \quad \underset{u \in \mathbb{R}^m}{\arg\min} \quad \|u\|_2^2 \tag{4a}$$
$$\text{s.t.} \quad L_f V(x) + L_g V(x)u + \lambda V(x) \leq 0. \tag{4b}$$

The above QP defines a min-norm feedback control law $u^* \colon \mathcal{X} \to \mathbb{R}^m$ that renders the origin exponentially stable.

B. Control Barrier Functions

Definition 2. Consider a continuously differentiable function $B \colon \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ and a set $\mathcal{C}$ defined as the superlevel set of B, $\mathcal{C} = \{x \in \mathcal{X} : B(x) \geq 0\}$. B is a *Control Barrier Function* (CBF) for system (1) if there exists an extended class $\mathcal{K}_\infty$ function γ such that for all $x \in \mathcal{X}$,
$$\sup_{u \in \mathbb{R}^m} \underbrace{L_f B(x) + L_g B(x)u}_{=\dot{B}(x,u)} + \gamma(B(x)) \geq 0. \tag{5}$$

If B is a CBF for system (1) and $\nabla B(x) \neq 0$ for all $x \in \partial\mathcal{C}$, any Lipschitz continuous control input u satisfying (5) renders the set $\mathcal{C}$ forward invariant [6, Thm. 2]. In [5], a QP formulation of a safety-critical controller is proposed by incorporating both conditions (3) and (5) as constraints:

CBF-CLF-QP:
$$u^*(x) = \quad \underset{(u,d) \in \mathbb{R}^{m+1}}{\arg\min} \quad \|u\|_2^2 + pd^2 \tag{6a}$$
$$\text{s.t.} \quad L_f V(x) + L_g V(x)u + \lambda V(x) \leq d, \tag{6b}$$
$$L_f B(x) + L_g B(x)u + \gamma(B(x)) \geq 0, \tag{6c}$$

where d is a slack variable used to relax the CLF constraint in order to give preference to safety over stability in case of conflict. In this paper, we will refer to (6b) as the *(relaxed) CLF constraint* and to (6c) as the *CBF constraint*. Note that the Lie derivatives of V and B, which appear in the these constraints, require explicit knowledge of the dynamics of the plant.

C. Adverse Effects of Model Uncertainty

We now provide some necessary settings and assumptions for our problem formulation. Let's assume that we have a *nominal model*:
$$\dot{x} = \tilde{f}(x) + \tilde{g}(x)u, \tag{7}$$
where $\tilde{f}$ and $\tilde{g}$ are Lipschitz continuous vector fields and $\tilde{f}(0) = 0$. In general, the nominal model vector fields $(\tilde{f}, \tilde{g})$ do not perfectly match the true plant vector fields (f, g) due to model uncertainty.

We start by designing a locally exponentially stabilizing CLF V and a CBF B, based on the nominal model (7). These functions are assumed to be a locally exponentially stabilizing CLF and a CBF, respectively, also for the true plant (1). This is a structural assumption, and it is met for feedback linearizable systems if the nominal model has the same degree of actuation as the true plant. A more detailed explanation can be found in [4]. Finally, we also assume that we can measure the state x and input u.

The main objective of the paper is to learn the CLF (6b) and CBF (6c) constraints for the true plant, in order to be able to synthesize safe and stabilizing controllers. Note that the actual derivatives of V and B depend on the true plant dynamics:
$$\dot{V}(x,u) = L_f V(x) + L_g V(x)u, \quad \dot{B}(x,u) = L_f B(x) + L_g B(x)u.$$

However, the nominal model-based estimates of their values are given by:
$$\tilde{\dot{V}}(x,u) = L_{\tilde{f}} V(x) + L_{\tilde{g}} V(x)u, \quad \tilde{\dot{B}}(x,u) = L_{\tilde{f}} B(x) + L_{\tilde{g}} B(x)u,$$

and can differ from the true values. We define Δ_V, Δ_B : $\mathcal{X} \times \mathbb{R}^m \to \mathbb{R}$ as the errors of the nominal model-based estimates:

$$\Delta_V(x,u) := \dot{V}(x,u) - \tilde{\dot{V}}(x,u), \tag{8}$$

$$\Delta_B(x,u) := \dot{B}(x,u) - \tilde{\dot{B}}(x,u). \tag{9}$$

Then, the CLF (6b) and CBF (6c) constraints become

$$L_{\tilde{f}}V(x) + L_{\tilde{g}}V(x)u + \Delta_V(x,u) + \lambda V(x) \leq d, \tag{10}$$

$$L_{\tilde{f}}B(x) + L_{\tilde{g}}B(x)u + \Delta_B(x,u) + \gamma(B(x)) \geq 0. \tag{11}$$

Therefore, by learning the uncertainty terms $\Delta_V(x,u)$ and $\Delta_B(x,u)$ correctly, we can have a good representation of the true CLF and CBF constraints, which can be used to construct a safe stabilizing controller for the true plant. Note that the uncertainty terms $\Delta_V(x,u)$ and $\Delta_B(x,u)$ are scalar values, leading to a lower dimensional learning problem than if we learned the dynamics f, g. We can approximately measure $\Delta_V(x,u)$ and $\Delta_B(x,u)$ by collecting trajectories from the true plant. These measurements are given by

$$z_j^V = (V(x(t+\Delta t)) - V(x(t)))/\Delta t - \tilde{\dot{V}}(x_j, u_j), \tag{12}$$

$$z_j^B = (B(x(t+\Delta t)) - B(x(t)))/\Delta t - \tilde{\dot{B}}(x_j, u_j), \tag{13}$$

where $x_j = (x(t+\Delta t) + x(t))/2$ is the mean of the state between $[t, t+\Delta t]$, and u_j is the control input during the same interval; Δt is the sampling interval; z_j^V and z_j^B are approximate measurements of $\Delta_V(x_j, u_j)$ and $\Delta_B(x_j, u_j)$, for $j = 1, \ldots, N$. Based on these data, we can define the regression problems for Δ_V and Δ_B as supervised learning problems. We will specifically use GP regression.

We close this section by revealing the control-affine structures of Δ_V and Δ_B. In (8) and (9), if we express $\dot{V}$, $\dot{B}$ and $\tilde{\dot{V}}$, $\tilde{\dot{B}}$ with their respective Lie derivatives, we get

$$\Delta_V(x,u) = \Phi_V \begin{bmatrix} 1 \\ u \end{bmatrix} := (L_f V - L_{\tilde{f}} V)(x) + (L_g V - L_{\tilde{g}} V)(x)u, \tag{14}$$

$$\Delta_B(x,u) = \Phi_B \begin{bmatrix} 1 \\ u \end{bmatrix} := (L_f B - L_{\tilde{f}} B)(x) + (L_g B - L_{\tilde{g}} B)(x)u, \tag{15}$$

where $\Phi_V, \Phi_B \in \mathbb{R}^{1 \times (m+1)}$. To simplify notations, we will introduce $y := [1 \ u^T]^T$. Then, both Δ_V and Δ_B are obtained by taking a dot product between a $(m+1)$ dimensional vector field (Φ_V or Φ_B) and y. This property will be used to construct a suitable kernel structure for the GP regression in the next section.

III. GAUSSIAN PROCESS REGRESSION

This section gives an overview of Gaussian Processes and how they are used for regression in this paper. We utilize a compound kernel, which was introduced in our previous work [7], to exploit the affine nature of the problem.

A. Gaussian Processes

A Gaussian Process (GP) is a type of random process such that any finite collection of its samples always has a joint Gaussian distribution. It is characterized by two functions: a mean function $q : \mathcal{X} \to \mathbb{R}$ and a covariance function $k :$ $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the input domain of the process. We express the process as:

$$h(x) \sim \mathcal{GP}(q(x), k(x,x')), \tag{16}$$

where $h(\cdot)$ is a sample drawn from it, and for any $x, x' \in \mathcal{X}$

$$\mathbb{E}[h(x)] = q(x), \ \mathbb{E}[(h(x)-q(x))(h(x')-q(x'))] = k(x,x').$$

Since the joint distribution of $h(x)$ and $h(x')$ is Gaussian, the functions q and k fully characterize the process. Note that the covariance function k determines the covariance matrix of the joint Gaussian distribution of finite samples. Therefore, only those functions that lead to positive semidefinite covariance matrices can be used for k. If k satisfies this requirement, it is said to be a positive definite kernel [21].

Given a set of finite measurements of the form $\{(x_j, h(x_j) + \varepsilon_j)\}_{j=1}^N$, where $\varepsilon_j \sim \mathcal{N}(0, \sigma_n^2)$ is white measurement noise, and a query point x_*, a posterior distribution for $h(x_*)$ can be derived from the Gaussian distribution of $[h(x_1), \cdots, h(x_N), h(x_*)]^T$ conditioned on the measurements. This can be used as a prediction of $h(x_*)$ at a query point x_*, with mean and variance

$$\mu_* = \mathbf{z}^T (K + \sigma_n^2 I)^{-1} K_*^T, \tag{17}$$

$$\sigma_*^2 = k(x_*, x_*) - K_*(K + \sigma_n^2 I)^{-1} K_*^T, \tag{18}$$

where $K \in \mathbb{R}^{N \times N}$ is the Gram matrix whose $(i,j)^{th}$ element is $k(x_i, x_j)$, $K_* = [k(x_*, x_1), \cdots, k(x_*, x_N)] \in \mathbb{R}^N$, and $\mathbf{z} \in \mathbb{R}^N$ is the vector containing the output measurements $z_j := h(x_j) + \varepsilon_j$. Here, we used $q(x) \equiv 0$ as the mean function in (16). The fact that this closed form expression of the prediction is available makes GPs attractive for many regression problems [22].

B. GP Regression with the Affine Dot Product Kernel

Our goal is to use GP regression for $\Delta_V(x,u)$ and $\Delta_B(x,u)$. Note that the input domain is now $\mathcal{X} \times \mathbb{R}^{m+1}$ instead of $\mathcal{X}$, where $\mathbb{R}^{m+1}$ is the space of $y = [1 \ u^T]^T$. We define $\bar{\mathcal{X}} := \mathcal{X} \times \mathbb{R}^{m+1}$. Even though any positive definite kernel defined for $\bar{\mathcal{X}}$ can be used, we would like to exploit the fact that both Δ_V and Δ_B are affine in y. With this goal in mind, the following compound kernel structure is used for our problem.

Definition 3. *Affine Dot Product Compound Kernel:* Define $k_c : \bar{\mathcal{X}} \times \bar{\mathcal{X}} \to \mathbb{R}$ given by

$$k_c \left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix} \right) := y^T Diag([k_1(x,x'), \cdots, k_{m+1}(x,x')]) y' \tag{19}$$

as the *Affine Dot Product* (ADP) compound kernel of $(m+1)$ individual kernels $k_1, \ldots, k_{m+1} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ [7].

There are two main benefits of using this specific compound kernel. First, it captures the appropriate structure of the target functions, which results in a much better regression fit compared to using arbitrary kernels. Since the target functions are characterized by a dot product between the $(m+1)$ dimensional vector field (Φ_V or Φ_B) and y, each individual kernel k_i, for $i = 1, \ldots, m+1$, represents an underlying kernel for the i-th element of the vector field.

The second benefit of using the ADP compound kernel is that it results in expressions for the mean and variance of the GP regression that are linear and quadratic in the control input, respectively. This is crucial for the construction of the convex min-norm controller that will be introduced in the next section. Let $X \in \mathbb{R}^{n \times N}$ and $Y \in \mathbb{R}^{(m+1) \times N}$ be matrices whose column vectors are the inputs x_j and y_j of the collected data, respectively. Then, when using the ADP compound kernel, Equations (17) and (18) give the following expressions for the mean and variance of the GP prediction at a query point (x_*, y_*):

$$\mu_* = \underbrace{\mathbf{z}^T (K_c + \sigma_n^2 I)^{-1} K_{*Y}^T}_{=:b_*^T} y_*, \qquad (20)$$

$$\sigma_*^2 = y_*^T \underbrace{\left(Diag\left(\begin{bmatrix} k_1(x_*, x_*) \\ \vdots \\ k_{m+1}(x_*, x_*) \end{bmatrix} \right) - K_{*Y}(K_c + \sigma_n^2 I)^{-1} K_{*Y}^T \right)}_{=:C_*} y_*. \qquad (21)$$

Here, $K_c \in \mathbb{R}^{N \times N}$ is the Gram matrix of k_c for the training data inputs (X, Y), and $K_{*Y} \in \mathbb{R}^{(m+1) \times N}$ is given by

$$K_{*Y} = \begin{bmatrix} K_{1*} \\ K_{2*} \\ \vdots \\ K_{(m+1)*} \end{bmatrix} \circ Y, \quad K_{i*} = [k_i(x_*, x_1), \cdots, k_i(x_*, x_N)].$$

C. Derivation of Confidence Bounds for Δ_V and Δ_B

In regression problems, the set of candidate functions, which is a design choice, determines the functions' expressivity at the cost of the complexity of the problem. In the machine learning literature, a Reproducing Kernel Hilbert Space (RKHS, [21]) is a common choice for this set since it encompasses a space of well-behaved and expressive-enough functions. An RKHS, denoted as $\mathcal{H}_k(\bar{\mathcal{X}})$, is characterized by a specific positive definite kernel k. The kernel k evaluates whether a member of $\mathcal{H}_k(\bar{\mathcal{X}})$ satisfies a specific property, namely a "reproducing" property: the inner product between any member $h \in \mathcal{H}_k(\bar{\mathcal{X}})$ and the kernel $k(\cdot, \bar{x})$ should reproduce h, i.e., $\langle h(\cdot), k(\cdot, \bar{x}) \rangle_k = h(\bar{x})$, $\forall \bar{x} \in \bar{\mathcal{X}}$. Furthermore, the RKHS norm $\|h\|_k := \sqrt{\langle h, h \rangle_k}$ is a measure of how "well-behaved" [1] the function $h \in \mathcal{H}_k(\bar{\mathcal{X}})$ is.

It turns out that if a target function h is a member of $\mathcal{H}_k(\bar{\mathcal{X}})$ with bounded RKHS norm, then a confidence bound for the true value of $h(\bar{x})$ can be specified by the mean and variance of the GP prediction [23]. We can apply this confidence bound analysis to GP regression problems that use the ADP compound kernel k_c (19) as the reproducing kernel. Let Δ be either Δ_V or Δ_B, and Φ be either Φ_V or Φ_B from (14) and (15). Δ is a dot product between Φ and $y = [1 \ u^T]^T$. The following theorem provides a probabilistic bound on the estimation error of Δ from the collected data.

Theorem 1. [7, Thm. 2] Consider $m+1$ bounded kernels k_i, for $i = 1, \ldots, (m+1)$. Assume that each i-th element of Φ is a member of $\mathcal{H}_{k_i}$ with bounded RKHS norm. We also assume that we have access to measurements $z_i = \Delta(x_i, u_i) + \varepsilon_i$, and

[1] $\|h(\bar{x}) - h(\bar{x}')\|_2 \le \|h\|_k \|k(\bar{x}, \cdot) - k(\bar{x}', \cdot)\|_k \ \forall \bar{x}, \bar{x}' \in \bar{\mathcal{X}}$

that each noise term ε_i is zero-mean and uniformly bounded by σ_n. Let $\beta := \left(2\eta^2 + 300\kappa_{N+1} \ln^3((N+1)/\delta) \right)^{0.5}$, with N the number of data points, η the bound of $\|\Delta\|_{k_c}$, and κ_{N+1} the maximum information gain after getting $N+1$ data points. Let μ_* and σ_*^2 be the mean (20) and variance (21) of the GP regression for Δ, using the ADP compound kernel k_c of $k_1, \cdots, k_{m+1}$, at a query point $(x_*, \ y_* = [1 \ u_*^T]^T)$, where y_* is an element of a bounded set $\mathcal{Y} \subset \mathbb{R}^{m+1}$. Then, with a probability of at least $1 - \delta$ the following holds for all $N \ge 1$, $x_* \in \mathcal{X}$, $[1 \ u_*^T]^T \in \mathcal{Y}$:

$$|\mu_* - \Delta(x_*, u_*)| \le \beta \sigma_*. \qquad (22)$$

For technical details of Theorem 1, please refer to [23, Thm. 6], and our previous paper [7].

IV. PROPOSED SAFETY CONTROLLER

In this section, we make use of the probability bounds given by Theorem 1 to build model uncertainty-aware CBF and CLF chance constraints that can be incorporated in a min-norm optimization problem.

From Theorem 1, each of the following inequalities hold with a probability of at least $1 - \delta$ for $\forall x \in \mathcal{X}$, $[1 \ u^T]^T \in \mathcal{Y}$:

$$\dot{V}(x, u) \le \tilde{V}(x, u) + \mu_V(x, u) + \beta \sigma_V(x, u), \qquad (23)$$

$$\dot{B}(x, u) \ge \tilde{B}(x, u) + \mu_B(x, u) - \beta \sigma_B(x, u), \qquad (24)$$

where $\mu_V(x, u)$, $\sigma_V(x, u)$ are the mean and standard deviation of the GP prediction of Δ_V at the query point (x, u), and $\mu_B(x, u)$, $\sigma_B(x, u)$ are the mean and standard deviation of the GP prediction of Δ_B at the query point (x, u), given by Equations (20) and (21).

In our previous work [7], we construct a Second-Order Cone Program (SOCP) using (23) to obtain an exponential CLF chance constraint. This SOCP defines a pointwise min-norm stabilizing feedback control law $u^*: \mathbb{R}^n \to \mathbb{R}^m$:

GP-CLF-SOCP:

$$u^*(x) = \quad \underset{u \in \mathbb{R}^m}{\arg \min} \|u\|_2^2 \qquad (25)$$

$$\text{s.t.} \quad \tilde{V}(x, u) + \mu_V(x, u) + \beta \sigma_V(x, u) + \lambda V(x) \le 0.$$

In this paper, we additionally incorporate a CBF chance constraint using the probabilistic bound on the CBF derivative given by (24). For that, we substitute this bound into the CBF constraint of (6c), resulting in a chance constraint that, if feasible, guarantees that the true CBF constraint of (6c) is satisfied with a probability of $1 - \delta$. This chance constraint is incorporated into the following optimization problem, which gives a safety-critical stabilizing feedback controller that takes into account the learned model uncertainty:

GP-CBF-CLF-SOCP:

$$u^*(x) = \underset{(u,d) \in \mathbb{R}^{m+1}}{\arg \min} \|u\|_2^2 + p \ d^2 \qquad (26a)$$

$$\text{s.t.} \quad \tilde{V}(x, u) + \mu_V(x, u) + \beta \sigma_V(x, u) + \lambda V(x) \le d, \quad (26b)$$

$$\tilde{B}(x, u) + \mu_B(x, u) - \beta \sigma_B(x, u) + \gamma(B(x)) \ge 0. \ (26c)$$

The above optimization problem is an SOCP if the ADP compound kernel of Definition 3 is used for both Δ_V and Δ_B [7, Thm. 3], thanks to the affine and quadratic structures of the means and variances, respectively. Also, it can be expressed in the standard form of SOCPs by transforming the quadratic objective into a Second-Order Cone (SOC) constraint and a linear objective, since the two chance constraints are SOC constraints.

This program is solved in real time to obtain a safe stabilizing control input at the current state. Note that the CLF constraint is relaxed in order to give preference to safety over stability in case of conflict. Also, the GP-CBF-CLF-SOCP does not need knowledge about the true plant dynamics, since it uses the estimates of Δ_V and Δ_B obtained from the GP regression.

V. ANALYSIS OF POINTWISE FEASIBILITY

For the feasibility analysis of the GP-CBF-CLF-SOCP, we only have to consider the single hard constraint of the problem, which is the CBF chance constraint (26c),

$$L_{\tilde{f}}B(x)+L_{\tilde{g}}B(x)u+\mu_B(x,u)-\beta\sigma_B(x,u)+\gamma(B(x)) \geq 0.$$

Remark 1. Unlike QPs, SOCPs with a single hard constraint can be infeasible. In this case, the problem will be infeasible if the uncertainty is high enough to prevent the existence of a control input that guarantees the system's safety at the current state with the desired probability of $1-\delta$. Therefore, by studying the conditions under which the GP-CBF-CLF-SOCP is feasible, we are actually trying to answer the question of how accurately we need to know our system in order to guarantee safety with a certain probability. This emerges as an important question for learning-based systems, as analyzed in recent work [24] for the stability of closed-loop systems.

From (20) and (21) we can obtain the mean μ_B and variance σ_B^2 of the GP prediction of Δ_B:

$$\mu_B(x,u)=b_B(x)^T\begin{bmatrix}1\\u\end{bmatrix}, \quad \sigma_B^2(x,u)=[1\ \ u^T]C_B(x)\begin{bmatrix}1\\u\end{bmatrix}. \tag{27}$$

We are using b_B and C_B instead of b_* and C_* to make clear that we are referring to the prediction of the CBF uncertainty Δ_B. Note that when k is a valid kernel and $\sigma_n > 0$, the Gram matrix of the variance, $C_B(\cdot)$ is always positive definite. We can therefore write

$$\sigma_B(x,u) = \left\| G(x)\begin{bmatrix}1\\u\end{bmatrix}\right\|, \tag{28}$$

where $G(\cdot) \in \mathbb{R}^{(m+1)\times(m+1)}$ is the matrix square root of $C_B(\cdot)$. Rewriting (26c) in terms of (27) and (28) gives

$$\beta\left\|G\begin{bmatrix}1\\u\end{bmatrix}\right\| \leq L_{\tilde{f}}B(x)+b_B(x)_1+\gamma(B(x))+L_{\tilde{g}}B(x)u+b_B(x)_{2:(m+1)}^Tu. \tag{29}$$

The numerical subscripts are used to indicate elements of vectors or columns of matrices. The inequality (29) can be easily expressed in the standard form of SOC constraints:

$$\|Q(x)u + r(x)\| \leq w(x)u + v(x), \tag{30}$$

with $Q := \beta G_{2:(m+1)} \in \mathbb{R}^{(m+1)\times m}$, $r := \beta G_1 \in \mathbb{R}^{(m+1)\times 1}$,

$$w(x) = \widehat{L_gB}(x) := L_{\tilde{g}}B(x) + b_B(x)_{2:(m+1)}^T \in \mathbb{R}^{1\times m}, \tag{31}$$

$$v(x) = \widehat{L_fB}(x) + \gamma(B(x)) \in \mathbb{R}, \tag{32}$$

where $\widehat{L_fB}(x) := L_{\tilde{f}}B(x) + b_B(x)_1$. $\widehat{L_gB}(x)$ is the mean of the prediction of $L_gB(x)$, as it incorporates the model-based term $L_{\tilde{g}}B(x)$ and the mean of the GP prediction of the actuation mismatch $b_B(x)_{2:(m+1)}^T$. Similarly, $\widehat{L_fB}(x)$ is the mean of the prediction of $L_fB(x)$. Note that

$$[v(x)\ w(x)]\begin{bmatrix}1\\u\end{bmatrix} = \widehat{L_fB}(x) + \widehat{L_gB}(x)u + \gamma(B(x)) \tag{33}$$

is the mean prediction of the left-hand side of the true plant CBF constraint (6c).

We now provide an equivalent formulation of constraint (30), which will be useful for the feasibility analysis.

Lemma 1. The SOC constraint (30) is feasible at $x \in \mathbb{R}^n$ if and only if there exists a $u \in \mathbb{R}^m$ such that both of the following two conditions hold:

$$\begin{cases} [1\ u^T]H(x)\begin{bmatrix}1\\u\end{bmatrix} \leq 0, & (34a)\\[2mm] w(x)u + v(x) \geq 0, & (34b) \end{cases}$$

where the symmetric matrix $H(x)$ takes the form

$$H(x)=\begin{bmatrix} r(x)^Tr(x)-v(x)^2 & r(x)^TQ(x)-v(x)w(x)\\ Q(x)^Tr(x)-w(x)^Tv(x) & Q(x)^TQ(x)-w(x)^Tw(x) \end{bmatrix}. \tag{35}$$

Proof. Inequality (34a) is obtained by simply taking squares on both sides of equation (30), and (34b) checks that the value of the right-hand side of (30) is non-negative, since the left-hand side is always non-negative. $\square$

A. Necessary Condition

With Lemma 1, we can now formulate a necessary condition for pointwise feasibility of the GP-CBF-CLF-SOCP.

Lemma 2 (Necessary condition for pointwise feasibility)**.** If the GP-CBF-CLF-SOCP of (26) is feasible at a point $x \in \mathbb{R}^n$, then the following condition must be satisfied:

$$[v(x)\ w(x)]\,C_B(x)^{-1}\begin{bmatrix}v(x)\\w(x)^T\end{bmatrix} \geq \beta^2, \tag{36}$$

or, equivalently, $H(x)$ cannot be positive definite.

Proof. Positive definiteness of $H(x)$ indicates that there does not exist any $u \in \mathbb{R}^m$ such that $[1\ u^T]H(x)\begin{bmatrix}1\\u\end{bmatrix} \leq 0$, and this is a contradiction to Lemma 1. Therefore, $H(x)$ cannot be positive definite if the GP-CBF-CLF-SCOP is feasible.

The rest of the proof shows that condition (36) is equivalent to $H(x)$ not being positive definite. Let $\psi(x) := [v(x)\ w(x)]$. Condition (36) does not hold if and only if

$$1 - \psi(x)\frac{1}{\beta^2}C_B(x)^{-1}\psi(x)^T = M/(\beta^2C_B(x)) > 0, \tag{37}$$

where $M = \begin{bmatrix}1 & \psi(x)\\\psi(x)^T & \beta^2C_B(x)\end{bmatrix}$. We use the operator $/$ for the Schur complement. From [25, Thm. 1.12], since $C_B(x)$

is positive definite, (37) holds if and only if M is positive definite. Applying again the same theorem, but this time to $M/1$, (37) is equivalent to $M/1 = \beta^2 C_B(x) - \psi(x)^T \psi(x) = [\mathrm{r}(x)\ Q(x)]^T [\mathrm{r}(x)\ Q(x)] - \psi(x)^T \psi(x) = H(x)$ being positive definite. Therefore, (36) holds if and only if $H(x)$ is not positive definite. □

The importance of Lemma 2 is that condition (36) encodes a trade-off between the values of the vector $[\mathrm{v}(x)\ \mathrm{w}(x)]$, information about the mean prediction of the CBF constraint as (33) indicates, and the uncertainty matrix $C_B(x)$.

Let us consider the one-dimensional input case for simplicity. The first trade-off is between $\mathrm{v}(x) = \widehat{L_f B}(x) + \gamma(B(x))$ and the upper left block of $C_B(x)$. Basically, this compares the prediction uncertainty σ_B with the violation of the mean prediction of the CBF constraint, when $u = 0$. The second trade-off is between $\mathrm{w}(x) = \widehat{L_g B}(x)$ and the lower right block of $C_B(x)$. If the value of $\mathrm{w}(x)$ is high, then the control input has a lot of influence on the resulting safety condition. Speaking informally, this means that the dynamics of $B(x)$ would be "highly controllable". For such case, a higher uncertainty in the input effects (reflected in the lower right block of $C_B(x)$) is allowed. To put it differently, if the "controllability" of $B(x)$ defeats the rate of growth of the actuation uncertainty as the norm of u increases, the SOCP gets feasible.

Remark 2. For the multi-dimensional input case, the above analysis hints towards the feasible input directions satisfying this trade-off between the "controllability" of $B(x)$ and the actuation uncertainty with respect to u. This interpretation leads to a more restrictive condition than (36), which is directly related to the sufficient condition for pointwise feasibility that will be introduced next.

Note that (36) checks the compromise between the full vector $[\mathrm{v}(x)\ \mathrm{w}(x)]$ and the uncertainty matrix C_B, which is a combination of both of the trade-off cases explained above.

B. Sufficient Condition

We now formulate a sufficient condition for feasibility of the GP-CBF-CLF-SOCP.

Lemma 3 (Sufficient condition for pointwise feasibility). For a point $x \in \mathbb{R}^n$, let $\lambda_\dagger$ be the minimum eigenvalue of the symmetric matrix $Q(x)^T Q(x) - \mathrm{w}(x)^T \mathrm{w}(x)$. If $\lambda_\dagger < 0$, the GP-CBF-CLF-SOCP (26) is feasible at x.

Proof. For a specific $x \in \mathbb{R}^n$, define
$$F := Q(x)^T Q(x) - \mathrm{w}(x)^T \mathrm{w}(x). \tag{38}$$
Let $e_\dagger$ be the unit eigenvector of F associated with the eigenvalue $\lambda_\dagger$. Then, $\lambda_\dagger < 0 \Rightarrow e_\dagger^T F e_\dagger < 0$. Since $Q(x)^T Q(x)$ is positive definite, this indicates that $\mathrm{w}(x) e_\dagger \neq 0$. Let's take a control input in the direction of this eigenvector:
$$u = \alpha \cdot \mathrm{sgn}(\mathrm{w}(x) e_\dagger) \cdot e_\dagger, \quad \alpha > 0. \tag{39}$$
Using this control input, the left-hand side of equation (34a) becomes $(\mathrm{r}(x)^T \mathrm{r}(x) - \mathrm{v}(x)^2) + 2\alpha \mathrm{sgn}(\mathrm{w}(x) e_\dagger)(\mathrm{r}(x)^T Q(x) - \mathrm{v}(x)\mathrm{w}(x)) e_\dagger + \alpha^2 e_\dagger^T F e_\dagger$, which can be made negative by choosing a large enough constant α, since $e_\dagger^T F e_\dagger < 0$.
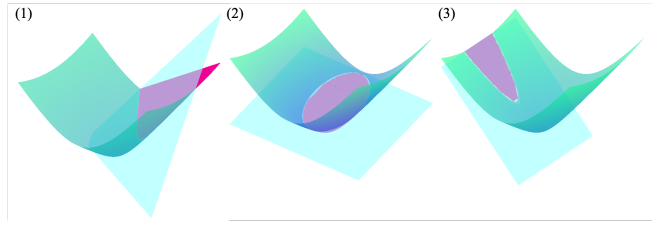

Fig. 1. Geometric interpretation of the feasibility conditions of the second-order cone constraint (Theorem 2). The green surface is the conic surface $\|Q(x)u + \mathrm{r}(x)\|$, the blue hyperplane is $(\mathrm{w}(x)u + \mathrm{v}(x))$, and the pink region is the feasible region. Case 1) hyperbolic intersection, Case 2) elliptical intersection, and Case 3) parabolic intersection. Note that for Cases 2) and 3), if (40) and (41) are not satisfied, respectively, the feasible region is empty.

Finally, plugging this control input into equation (34b) yields $|\mathrm{w}(x) e_\dagger|\, \alpha + \mathrm{v}(x) \geq 0$, which again holds for a sufficiently large α. Therefore, (26) is feasible by Lemma 1. □

This condition is directly related to Remark 2 in the sense that $\lambda_\dagger < 0$ means that there exists an input direction such that the "controllability" of $B(x)$ defeats the rate of growth of the actuation uncertainty as the norm of u increases.

The crux of this sufficient condition is that a single scalar value ($\lambda_\dagger$) being negative guarantees the feasibility of the problem, and this condition can be easily checked online before solving (26). Note that the value of $\lambda_\dagger$ is dependent on the distribution of data around the current state $x \in \mathbb{R}^n$. We believe that this condition can be used for the design of online safe learning algorithms, and we plan to explore this direction in our future work.

C. Necessary and Sufficient Condition

Finally, we state the necessary and sufficient condition for pointwise feasibility. This condition includes those of Lemmas 2 and 3 with some additional cases.

Theorem 2 (Necessary and sufficient condition for pointwise feasibility). For a given state $x \in \mathbb{R}^n$, let $\lambda_\dagger$ be the minimum eigenvalue of the symmetric matrix F defined in (38). The GP-CBF-CLF-SOCP (26) is feasible at x if and only if (36) is satisfied and one of the following cases holds:

1) $\lambda_\dagger < 0$;
2) $\lambda_\dagger > 0$, and
$$\mathrm{v}(x) - \mathrm{w}(x) F^{-1} \{Q(x)^T \mathrm{r}(x) - \mathrm{w}(x)^T \mathrm{v}(x)\} \geq 0; \tag{40}$$
3) $\lambda_\dagger = 0$, and
$$\mathrm{v}(x) - \mathrm{w}(x) \left(Q(x)^T Q(x)\right)^{-1} Q(x)^T \mathrm{r}(x)^T > 0. \tag{41}$$

Case 1) is exactly Lemma 3, and it corresponds to the feasible set being hyperbolic. Cases 2) and 3) correspond to elliptic and parabolic feasible sets, respectively (Fig. 1).

Proof. See Appendix. □

We believe that Theorem 2 constitutes the first step towards understanding what conditions the distribution of data should satisfy in order to obtain probabilistic safety guarantees in the presence of actuation uncertainty. This problem is related to the notion of persistency of excitation [3]. However, we think that Theorem 2 hints at less restrictive conditions that are sufficient to achieve the desired
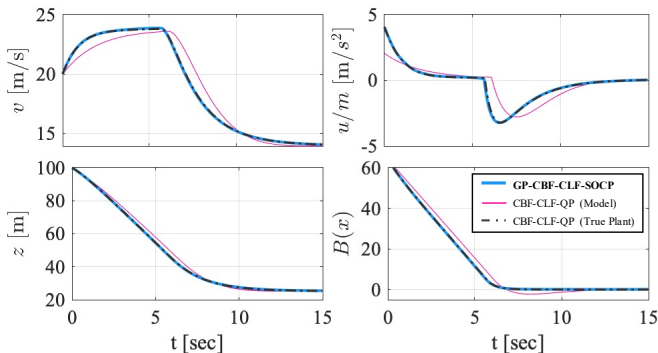
Fig. 2. Simulation results of applying the GP-CBF-CLF-SOCP (blue) to the adaptive cruise control example under model uncertainty, compared to the nominal-model-based CBF-CLF-QP (red) and the true-plant-based CBF-CLF-QP (grey). Note that the simulation ends with v decreased since it is approaching the front vehicle, whose speed (14 m/s) is lower than the desired speed (24 m/s).
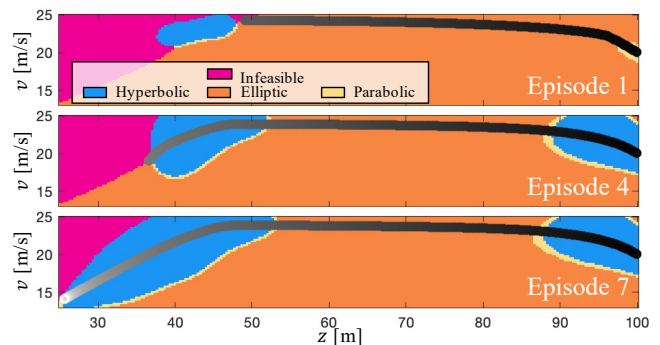

Fig. 3. Evolution of the feasible region during the learning episodes for the adaptive cruise control example. The trajectories evolve from black to white. As more data is collected, the feasible set gets larger. We ran a total of 7 episodes, and the final number of data points is $N = 189$.

safety performance. Verifying specific conditions on the data distribution remains as our future work. Finally, note that the presented analysis is also directly applicable to the GP-CLF-SOCP of (25) by considering the CLF chance constraint instead of the CBF one.

VI. NUMERICAL RESULTS

To validate our proposed controller we utilize the following model of an adaptive cruise control system:

$$\dot{x} = f(x) + g(x)u, \quad f(x) = \begin{bmatrix} -F_r(v)/m \\ v_0 - v \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ 1/m \end{bmatrix}, \quad (42)$$

where $x = [v \ z]^T \in \mathbb{R}^2$ is the system state, with v being the forward velocity of the ego car, and z the distance between the ego car and the front car. $u \in \mathbb{R}$ is the ego car's wheel force as control input, v_0 is the velocity of the front car which is assumed to be constant (14 m/s), m is the mass of the ego car, and $F_r(v) = f_0 + f_1 v + f_2 v^2$ is the rolling resistance acting on the ego car.

The control objective is to reach a desired speed command of $v_d = 24$ m/s while maintaining a safe distance $z \geq T_h v$ with respect to the front vehicle, where $T_h = 1.8$ s is the lookahead time. For this, we design a CLF as $V(x) = (v - v_d)^2$ and a CBF as $B(x) = z - T_h v$.

In order to introduce model uncertainty, for the nominal model we use $m = 1650$ kg, $f_0 = 0.1$, $f_1 = 5$, $f_2 = 0.25$; and for the true plant $m = 3300$ kg, $f_0 = 0.2$, $f_1 = 10$, $f_2 = 0.5$.

As it can be seen in the lower right plot of Fig. 2, a CBF-CLF-QP controller constructed based on the nominal model (depicted in pink) violates the safety constraint after about 7 seconds of simulation (since $B(x) < 0$). This is caused by the model uncertainty. In order to correct this, we use the GP regression approach that has been presented in this paper for both CLF and CBF constraints.

The data collection process is conducted in an episodic fashion. We start by running an initial rollout of the nominal CBF-CLF-QP until the system violates the safety constraint. With that data, we obtain a first GP model that is now used to run a rollout with the GP-CBF-CLF-SOCP. We stop the rollout if the optimization problem becomes infeasible or if the system becomes unsafe. Because of the high probability

bound $(1 - \delta = 0.95)$ we use for (23) and (24), infeasibility always occurs before the system becomes unsafe. We then add the collected data to the GP model and run the next rollout. This process is repeated until the rollout is completely feasible. Fig. 3 represents this process, and it also depicts how the feasibility region evolves as more data is collected.

In Fig. 2 the performance of the GP-CBF-CLF-SOCP controller of the final episode is shown in blue. For reference, the results of the simulation with the CBF-CLF-QP controller that uses the true plant dynamics are also illustrated in the plots. The proposed GP-CBF-CLF-SOCP closely matches the performance of the true plant-based CBF-CLF-QP and always respects safety, whereas the nominal model-based CBF-CLF-QP violates the safety constraint due to model uncertainty. Finally, the computation time of the SOCP per timestep was 7.4 ± 4.2 ms on a laptop with an Intel Core i7 and 32 GB of RAM, fast enough for real-time applications.

VII. CONCLUSION

We presented a framework to construct a GP-based safe stabilizing controller (GP-CBF-CLF-SOCP) that takes into account the problem of model uncertainty in the CLF and CBF constraints. We also presented an analysis of pointwise feasibility of the proposed GP-CBF-CLF-SOCP, which we believe should pave the way to obtain safety guarantees for systems with uncertain dynamics and input effects without requiring the infeasible collection of data that fully characterize the dynamics of the system. This will be the focus of our future work.

APPENDIX

Proof of Theorem 2
We start the proof by providing a geometric interpretation of the SOC constraint. $\|Q(x)u + r(x)\|$ is an m-dimensional conic section resulting from the intersection of the second-order cone $\|[r(x) \ Q(x)]y\|$ and the hyperplane $[1 \ 0 \ \cdots \ 0] \cdot y = 1$ defined on $\mathbb{R}^{m+1}$ space. As a reminder, $y = [1 \ u^T]^T$. In the input space, $\mathbb{R}^m$, we will call $\|Q(x)u + r(x)\|$ a conic

surface (illustrated in Fig. 1). This surface asymptotically converges to the cone $\|Q(x)(u - u_0)\|$ as $\|u\| \to \infty$, where

$$u_0 = -\left(Q(x)^T Q(x)\right)^{-1} Q(x)^T \mathrm{r}(x) \qquad (43)$$

is the u that minimizes $\|Q(x)u + \mathrm{r}(x)\|$, given by the least-squares solution. We will refer to the cone $\|Q(x)(u - u_0)\|$ as the asymptote of the conic surface $\|Q(x)u + \mathrm{r}(x)\|$.

Since $Q(x) \succ 0$, the problem (26) is feasible if and only if an intersection between the conic surface $\|Q(x)u + \mathrm{r}(x)\|$ and the hyperplane defined by $(\mathrm{w}(x)u + \mathrm{v}(x))$ exists.

Case 1) Lemma 3 is the proof. Note that this condition itself implies that (36) is satisfied. The slope of the hyperplane $(\mathrm{w}(x)u + \mathrm{v}(x))$ is greater than the slope of the asymptote of the conic surface (Fig. 1.(1)).

Case 2) Since the smallest eigenvalue of F is positive, this means that $F \succ 0$. Note that the lower right block of the matrix $H(x)$ in (35) is F. Therefore, $F \succ 0$ implies that the left-hand side of equation (34a) is strictly convex, attaining the global minimum at some $u = u_1 \in \mathbb{R}^m$. The first order optimality condition gives

$$u_1 = -F^{-1} h,$$

with $h := Q(x)^T \mathrm{r}(x) - \mathrm{w}(x)^T \mathrm{v}(x)$. Since at u_1 the minimum is achieved, equation (34a) is satisfied if and only if

$$[1 \; u_1^T] H(x) \begin{bmatrix} 1 \\ u_1 \end{bmatrix} \leq 0. \qquad (44)$$

Plugging (35) and $u_1 = -F^{-1} h$ into (44), we get

$$\left(\mathrm{r}(x)^T \mathrm{r}(x) - \mathrm{v}(x)^2\right) - h^T F^{-1} h = H(x)/F \leq 0. \qquad (45)$$

Since $H(x)$ cannot be positive definite by the necessary condition (36), and F is positive definite, from [25, Thm. 1.12] (45) must be satisfied. Consequently, (34a) holds for $u = u_1$. Now, plugging u_1 into (34b) we have: $\mathrm{w}(x)u_1 + \mathrm{v}(x) = \mathrm{v}(x) - \mathrm{w}(x)F^{-1}\{Q(x)^T \mathrm{r}(x) - \mathrm{w}(x)^T \mathrm{v}(x)\}$, which is precisely the left-hand side of (40). The feasible region is non-empty if and only if the above expression is greater than or equal to 0 due to Lemma 1 (Fig. 1.(2)). If the above expression is smaller than 0, it means that the hyperplane $(\mathrm{w}(x)u + \mathrm{v}(x))$ intersects with the negative conic surface, $-\|Q(x)u + \mathrm{r}(x)\|$, forming an ellipse, and cannot intersect with the positive conic surface. Therefore, when $\lambda_\dagger > 0$, the SOCP (26) is feasible if and only if (40) is satisfied.

Case 3) Note that $\lambda_\dagger = 0$ means that the hyperplane $(\mathrm{w}(x)u + \mathrm{v}(x))$ has the same slope as the asymptote of the conic surface $\|Q(x)u + \mathrm{r}(x)\|$ (Fig. 1.(3)). Define

$$p := \mathrm{v}(x) - \mathrm{w}(x)\left(Q(x)^T Q(x)\right)^{-1} Q(x)^T \mathrm{r}(x)^T. \qquad (46)$$

Then, the given condition (41) holds if and only if $p > 0$. Consider $u = u_0$ from (43) that minimizes $\|Q(x)u + \mathrm{r}(x)\|$. Then, $p = \mathrm{v}(x) + \mathrm{w}(x)u_0$. Let $e_\dagger$ be the unit eigenvector of F associated with the eigenvalue $\lambda_\dagger = 0$. Then, $e_\dagger^T F e_\dagger = 0$. Since $Q(x)^T Q(x)$ is positive definite, this means that $\mathrm{w}(x)e_\dagger \neq 0$. Now let's consider a control input of the form

$$u = u_0 + \alpha \cdot \mathrm{sgn}(\mathrm{w}(x)e_\dagger) \cdot e_\dagger, \quad \alpha > 0. \qquad (47)$$

Using this control law, the left-hand side of (34a) becomes

$$\{\mathrm{r}(x)^T \mathrm{r}(x) - \mathrm{v}(x)^2 + 2h^T u_0 + u_0^T F u_0\} - 2\alpha \cdot p \cdot |\mathrm{w}(x)e_\dagger|, \qquad (48)$$

and the left-hand side of (34b) becomes

$$p + \alpha \cdot |\mathrm{w}(x)e_\dagger|. \qquad (49)$$

When $p > 0$, there exists a sufficiently large α such that (48) becomes non-positive and (49) becomes positive. Therefore, from Lemma 1, the SOCP (26) is feasible.

Note that the geometric interpretation of the condition $p > 0$ is that the hyperplane $(\mathrm{w}(x)u + \mathrm{v}(x))$, which has the same slope as the asymptote of the conic surface $\|Q(x)u + \mathrm{r}(x)\|$ for this case, should be placed over the asymptote in order for it to intersect with the surface. At $u = u_0$, the asymptote $\|Q(x)(u - u_0)\|$ takes value 0. p is the value of the hyperplane at $u = u_0$. Therefore, when $p \leq 0$, the hyperplane is always under the conic surface, and never intersects it. Consequently, the constraint (30) is not feasible.

REFERENCES

[1] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *European Control Conference*, 2019, pp. 307–312.

[2] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning*. Citeseer, 2011, pp. 465–472.

[3] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[4] A. J. Taylor, V. D. Dorobantu, S. Dean, B. Recht, Y. Yue, and A. D. Ames, "Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty," *arXiv:2011.10730*, 2020.

[5] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.

[6] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, 2019, pp. 3420–3431.

[7] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics," in *American Control Conference*, 2021.

[8] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 7, pp. 1163 – 1173, 1983.

[9] Q. T. Nguyen, "Robust and adaptive dynamic walking of bipedal robots," Ph.D. dissertation, Carnegie Mellon University, 2017.

[10] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *American Control Conference*, 2020, pp. 1399–1405.

[11] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, 2021.

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

[13] J. Choi, F. Castañeda, C. Tomlin, and K. Sreenath, "Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions," in *Robotics: Science and Systems*, Corvalis, OR, 2020.

[14] T. Westenbroek, A. Agrawal, F. Castaneda, S. S. Sastry, and K. Sreenath, "Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers," in *IFAC Analysis and Design of Hybrid Systems (ADHS)*, Brussels, Belgium, July 2021.

[15] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*, 2020, pp. 708–717.

[16] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes," in *IEEE Conference on Decision and Control*, 2016, pp. 4661–4666.

[17] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A. a. Agha-mohammadi, and E. A. Theodorou, "Bayesian learning-based adaptive control for safety critical systems," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 4093–4099.

[18] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," *arXiv:2004.05273*, 2020.

[19] V. Dhiman, M. J. Khojasteh, M. Franceschetti, and N. Atanasov, "Control barriers in bayesian learning of system dynamics," *arXiv:2012.14964*, 2020.

[20] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs," in *Control of Cyber-Physical Systems*. Springer, 2013, pp. 219–240.

[21] H. Wendland, *Scattered data approximation*. Cambridge university press, 2004, vol. 17.

[22] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press, Cambridge, MA, 2006, vol. 2, no. 3.

[23] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *International Conference on Machine Learning*. Madison, WI, USA: Omnipress, 2010, p. 1015–1022.

[24] A. Lederer, A. Capone, T. Beckers, J. Umlauft, and S. Hirche, "The impact of data on the stability of learning-based control," in *Learning for Dynamics and Control*, 2021, pp. 623–635.

[25] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.