Constraint-Guided Online Data Selection for Scalable Data-Driven Safety Filters in Uncertain Robotic Systems

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

***Abstract*—As the use of autonomous robotic systems expands in tasks that are complex and challenging to model, the demand for robust data-driven control methods that can certify safety and stability in uncertain conditions is increasing. However, the practical implementation of these methods often faces scalability issues due to the growing amount of data points with system complexity, and a significant reliance on high-quality training data. In response to these challenges, this study presents a scalable data-driven controller that efficiently identifies and infers from the most informative data points for implementing data-driven safety filters. Our approach is grounded in the integration of a model-based certificate function-based method and Gaussian Process (GP) regression, reinforced by a novel online data selection algorithm that reduces time complexity from quadratic to linear relative to dataset size. Empirical evidence, gathered from successful real-world cart-pole swing-up experiments and simulated locomotion of a five-link bipedal robot, demonstrates the efficacy of our approach. Our findings reveal that our efficient online data selection algorithm, which strategically selects key data points, enhances the practicality and efficiency of data-driven certifying filters in complex robotic systems, significantly mitigating scalability concerns inherent in nonparametric learning-based control methods.**

***Index Terms*—Safety-critical systems, Learning-based control, Safety filters, Safety, Stability**

I. Introduction

A. Motivation

As autonomous robots become increasingly prevalent in our everyday lives, incidents such as fatal accidents involving self-driving cars have highlighted the importance of ensuring that robotic systems adhere to various system-critical constraints. Examples of these constraints include safety requirements for self-driving cars to prevent accidents, or stability conditions for legged robots to avoid falling over. Failure to meet these constraints can lead to catastrophic consequences.

Learning-based control methods have gained considerable attention in recent years due to their capacity to accomplish complex tasks by leveraging vast amounts of data. However, a fundamental challenge in deploying these emerging methods for real-world robots is ensuring their adherence to the considered system-critical constraints.

One way of addressing this challenge is through *a-posteriori verification*, i.e., analyzing the properties of these policies after they have been synthesized. A significant body of recent work on neural network verification follows this approach [1]. If the neural policy of interest does not pass the certification test, the designer typically modifies the learning setup and reiterates this process until a satisfying policy is found. This can be a time-consuming and computationally expensive process, although there are ongoing efforts in the community to overcome this challenge by using the result from verification to guide the control design process [2], [3].

An alternative to the a-posteriori certification approach is using a model-based *certifying filter*, supplemented by data to address the limitations imposed by imperfect models. In this scheme, a certifying filter is designed using available mathematical models to ensure compliance with system-critical constraints, and then the filter is enhanced by incorporating data from the actual system to address discrepancies arising from the imperfect model. This concept is often referred to as the *data-driven safety filter* [4].

The data-driven component of the filter learns how to address the effect of the discrepancy between the mathematical model and the actual system in choosing the control input that renders the system to satisfy the desired constraints. Many of these methods employ nonparametric learning techniques, such as Gaussian Process regression [5]. This approach is used because it provides not only predictions but also uncertainty estimates for those predictions, which can be used to quantify possible prediction errors of the learned model and maintain the desired level of safety despite any errors that the learned model may make.

However, there are two primary challenges associated with the data-driven certifying filter. First, nonparametric methods generally do not scale well with an increasing number of data points, which can significantly limit their application to more complex systems. As system complexity increases, characterizing the effect of the imperfect model requires more data, and controllers need to operate at higher frequencies to effectively manage rapidly changing system dynamics. Second, the success of a data-driven certifying filter in achieving its objective, like any other learning-based control approach, relies on the quality of the available training data [6], [7]. If the information derived from the data about the real system is insufficient, the data-driven controller can fail to adhere to the system-critical constraints.

These two challenges are tightly coupled; when a large dataset is available, it is crucial to investigate which data points are most critical for meeting the certifying filter's objective

*The first three authors contributed equally to the work. J. J. Choi, F. Castañeda, B. Zhang, C. J. Tomlin, and K. Sreenath are with the University of California, Berkeley, CA, 94720, USA. W. Jung is with Georgia Institute of Technology, GA, 30332, USA.

to properly address the scalability challenge. This raises the motivating question of the paper: by focusing on the most relevant data for system-critical constraints, can we extend the applicability of data-driven certifying filters to more complex and uncertain real-world robotic systems?

B. Contributions

In this paper, we introduce an efficient approach for determining the most relevant data points for deploying data-driven certifying filters on real-world robotic systems. Our method expands on our earlier work [8], [9], where we designed a data-driven filter that combines a model-based control method rooted in certificate functions, such as Control Barrier Functions (CBFs [10]) and Control Lyapunov Functions (CLFs [11]), with a Gaussian Process (GP) regression for the data-driven component. The filtered control input, guaranteed to satisfy system-critical constraints with high probability, is determined by solving a second-order cone program (SOCP). We delve into understanding which data points are critical for ensuring the feasibility of the SOCP filter and subsequently, for meeting the system-critical constraints. Guided by this understanding, we develop an efficient online data selection algorithm for the filter. Each time the SOCP controller is executed, this algorithm selects only the most relevant data points to secure the SOCP's feasibility, which considerably enhances the time complexity of the GP-based safety filter.

Using the proposed approach, we showcase the applicability of Gaussian Process-based safety filters to high-dimensional and real robotic systems handling large datasets, overcoming the scalability constraints that previously limited the use of such filters to simple toy systems [7]–[9], [12]–[16]. We demonstrate the successful deployment of our method in a real cart-pole experiment to ensure the cart remains within its position limit and a 10-dimensional bipedal robot in simulation that attempts stable walking while subjected to model errors.

C. Notations

B: binary correlation indicator matrix (32)
C: certificate function (Definition 1)
$\mathbb{D}_N := \{\bar{x}_j, \bar{z}_j\}_{j=1}^N$: entire dataset for GP regression
$\mathbb{D}_M (\subset \mathbb{D}_N)$: online dataset
f, g: true plant vector fields ((1))
$\tilde{f}, \tilde{g}$: nominal model vector fields ((2))
$F_{\mathbb{D}_M}$: objective function of the data selection algorithm ((26))
$k_f, k_{g_1} \cdots, k_{g_m}$: individual kernels (Definition 2)
$\mathbf{k}$: Affine Dot Product (ADP) kernel (Definition 2)
$\mathbf{k}^u$: ADP kernel that captures only the control vector field-relevant part ((21))
$K_{\mathbb{D}_N}, K_{\mathbb{D}_M}$: kernel matrix whose $(i,j)^{th}$ element is $k(\bar{x}_i, \bar{x}_j)$
$K_* := [k(\bar{x}_*, \bar{x}_1), \ \cdots \ , k(\bar{x}_*, \bar{x}_N)] \in \mathbb{R}^N$
K_{*U}: (13)
$\widehat{L_f C}(x|\mathbb{D}_{M,N})$, $\widehat{L_g C}(x|\mathbb{D}_{M,N})$: GP mean-based estimate of the Lie derivatives of C ((16))
m: control input dimension
M: number of online data points
n: state dimension
$n_i(x,u)$: kernel-based alignment measure ((27))
N: number of entire data points
u: control input
u_{ref}: reference controller
U: control input bound
x: state
$\bar{x} := (x,u)$: input for GP regression
$\mathcal{X}$: state domain
$\bar{\mathcal{X}} = \mathcal{X} \times \mathbb{R}^m$: GP input domain
$\bar{z}_j$: noisy measurement of Δ at query $\bar{x}_j$
$\mathbf{z}$: vector consisting of the dataset outputs, $\bar{z}_j$
β: constant in Assumption 1
γ: comparision function in Definition 1
δ: probability level in Assumption 1
Δ: model uncertainty term ((7))
ϵ: constant in Theorem 1
$\mu(x,u|\mathbb{D}_{M,N})$: GP posterior mean
$\mathcal{M}(x|\mathbb{D}_N)$: (11)
$\sigma^2(x,u|\mathbb{D}_{M,N})$: GP posterior variance
σ_n^2: measurement noise variance
$\Sigma(x|\mathbb{D}_N)$: Gram matrix of GP posterior variance ((12))
$\boldsymbol{\varphi}(x,u)$: ADP kernel's feature vector

II. Related Work

Control Barrier Functions (CBFs, [10]) and Control Lyapunov Functions (CLFs, [11]) are model-based certificate functions that can be used to design policy filters to enforce safety and stability, respectively, of a controlled system. While initially conceived for systems with perfectly known dynamics, early results showed how to extend these filters to robust [17]–[20] and adaptive [21]–[23] control settings to address the issue of imperfect models.

The integration of these certificate filters with data-driven methods has become increasingly popular for systems with uncertain dynamics. Several studies employ neural networks to learn the model mismatch terms [24]–[26]. Despite their practicality and effectiveness, verifying the accuracy of neural network predictions can be challenging.

Alternative approaches, upon which our work builds, use nonparametric regression techniques for this purpose [7]–[9], [12]–[16], [27]. Most notably, Gaussian Process (GP) regression models provide a probabilistic assurance of prediction quality under mild assumptions [28], [29].

The GP research community has a rich history in developing methods to improve the computational complexity of GP inference, commonly referred to as Sparse GP regression [30], [31]. The work in [27] uses one of this methods (random features approximation) to speed up GP inference for data-driven safety filters. Additionally, existing approaches that quantify the importance of data for system identification mostly focus on optimizing information-theoretic metrics, such as the information gain, when developing exploration strategies [32]–[37]. However, these general-purpose methods lack awareness of any control objective. Instead of aiming to obtain an approximate global GP regression model, the method we introduce in this paper utilizes certificate functions to select a small set of data points, online at each state, that are useful for certification.

However, obtaining the best subset of data constitutes a combinatorial optimization problem that would be more computationally demanding than performing exact GP inference. For this reason, we instead present a control-informed efficient approximate data selection method that effectively serves to reduce the inference time of data-driven safety filters. This enables the deployment of these filters on real robotic systems.

The authors of [6], [38] propose a method to evaluate the importance of data for maintaining the stability of data-driven closed-loop systems. As such, they study the connection between data and the performance of a particular given policy. Additionally, they introduce a greedy data selection strategy for GP inference based on an importance measure they propose. However, these selection strategies are still too computationally expensive to run online. Our work instead tackles the problem of robust control design, studying online which data is most relevant to achieve a desired certification property in the resulting data-driven control policy. Furthermore, our approach characterizes the relationship between data and safety in the control input space, emphasizing the richness of each data point for the specific certification objective, rather than relying on data density measures. This is a similar objective to the one of [39], where an algorithm to select the most useful data points for successfully performing multiple control tasks is presented. However, this method also suffers from scalability issues that prevent it from being applicable to real robotic systems.

III. Certifying Filters for Uncertain Systems

A. Uncertain Dynamics and Certifying Filter

In this paper, we examine a control-affine system as shown below:

$$\dot{x} = f(x) + g(x)u. \tag{1}$$

Here, $x \in \mathcal{X} \subset \mathbb{R}^n$ represents the state, and $u \in \mathbb{R}^m$ denotes the control input. This form is suitable for representing various robotic systems, including those with Lagrangian dynamics. We assume that both f and g are locally Lipschitz continuous, and without loss of generality, we consider $f(0) = 0$ so that $x = 0$ is an equilibrium. Throughout the paper, we will refer to the system described in (1) as the *true plant*.

This paper addresses the challenge of ensuring critical system constraints for the true plant (1), such as safety and stability, when its dynamics f and g are unknown, while trying to accomplish a desired task. We assume that a controller for the desired task has already been designed and is provided as a *reference controller* $u_{\text{ref}} : \mathcal{X} \to \mathbb{R}^m$. In the absence of the reference controller, we can consider $u_{\text{ref}}(x) \equiv 0$. This controller is often unaware of the system's constraints that are vital for preventing catastrophic failure, which we refer to as *system-critical constraints*. Common examples of these constraints include safety constraints, which can be expressed as constraints in the system's state space, and stability constraints that maintain the system's stability around a desired equilibrium point.

We aim to design a *certifying filter* that operates between the reference controller u_{ref} and the true plant, ensuring the control applied to the true plant is filtered to satisfy the relevant system-critical constraint. When the reference controller u_{ref} adheres to the constraint, the certifying filter simply passes $u_{\text{ref}}(x)$ to the true plant. However, if u_{ref} violates the constraint, the filter minimally overrides it with a safe control signal to prevent system failure. This filtering structure is known by various names, most notably as a *safety filter* [4], [40]. Since this structure decouples the design process for safety assurance from the design procedure for achieving performance and thus, reducing the complexity of the control system design, it has been demonstrated to be an effective control architecture for numerous real-world applications [41].

For this purpose, we assume access to an approximate *nominal model* of the true plant's dynamics, represented by $\tilde{f} : \mathcal{X} \to \mathbb{R}^n$ and $\tilde{g} : \mathcal{X} \to \mathbb{R}^{n\times m}$:

$$\dot{x} = \tilde{f}(x) + \tilde{g}(x)u. \tag{2}$$

Consequently, the true plant dynamics in (1) are uncertain, and only a nominal model (2) is available. This nominal model serves as the starting point for the design steps of the certifying filter, which will be discussed subsequently, and represents the designer's best estimate of the true plant. The degree of accuracy required for approximating the true plant with the nominal model, in the design steps we undertake, will be discussed in Remark 1.

B. Certificate Function-based Design

A vital step in designing the certifying filter involves utilizing the concept of *certificate functions* [42], which are also known by various names, such as safety index in [43] or energy function in [44]. Informally, a certificate function is a scalar function of the state, and its value and gradient can be used to establish a sufficient condition for a control input u to satisfy the desired system-critical constraint. This condition can then be employed as a *certifying constraint* in the certifying filter for the control input. If $u_{\text{ref}}(x)$ fails to meet the constraint, it is overridden with an appropriate control input u that satisfies the constraint. The idea of using such scalar functions traces back to Lyapunov functions, energy-like functions that certify the stability of an equilibrium [45].

In this paper, we focus on Control Barrier Functions (CBFs) [10] and Control Lyapunov Functions (CLFs) [11] as specific examples of certificate functions, since they are the most prevalent choices for ensuring the satisfaction of safety and stability constraints in a system, respectively [42]. We put forth a definition of a certificate function that unifies both CBFs and CLFs under a single definition, as similarly proposed in [12]:

Definition 1. A function $C : \mathcal{X} \to \mathbb{R}$ is a *certificate function* for the true plant (1) with an extended class $\mathcal{K}_\infty$ function $\gamma : \mathbb{R} \to \mathbb{R}$ (called comparison function) if

1) for all $x \in \mathcal{X}$, there exists $u \in \mathbb{R}^m$ such that

$$\dot{C}(x,u) + \gamma(C(x)) \geq 0, \tag{3}$$

where $\dot{C}(x,u)$ is the Lie derivative of C for the true plant (1), that is,

$$\dot{C}(x,u) = \underbrace{\nabla C(x) \cdot f(x)}_{L_f C(x)} + \underbrace{\nabla C(x) \cdot g(x)}_{L_g C(x)} u, \tag{4}$$

2) and if $u(t)$ satisfying (3) for all $t \geq 0$ is a sufficient condition for $x(t)$ satisfying the desired system-critical constraint for all $t \geq 0$.

The system-critical constraint for the CBF is that the trajectory stays inside the zero-superlevel set of C indefinitely, i.e., $x(t) \in \mathcal{C} := \{x \in X \mid C(x) \geq 0\}$ for all $t \geq 0$ [10]. The

system-critical constraint for the CLF is that the trajectory is asymptotically stable to the equilibrium $x = 0$ [46]. Both CBFs and CLFs satisfy the aforementioned definition of certificate functions. Note that to align with the inequality form in (3), we need to negate the CLF. This adjustment ensures that both CBFs and CLFs can be used within the same framework to satisfy the desired system-critical constraints.

Given a reference controller $u_{\text{ref}} : \mathcal{X} \to \mathbb{R}^m$, the condition in (3) can be used to formulate a minimally-invasive certifying filter [10]:

Certificate Function-based Quadratic Program (CF-QP):

$$u^*(x) = \quad \underset{u\in\mathbb{R}^m}{\arg\min} \quad \|u - u_{\text{ref}}(x)\|_2^2 \tag{5a}$$

$$\text{s.t.} \quad L_f C(x) + L_g C(x) u + \gamma(C(x)) \geq 0. \tag{5b}$$

It is important to note that the constraint (5b) is affine in u, which means that the optimization problem is a quadratic program (QP). This problem is solved pointwise in time to obtain a filtered control law $u^* : \mathcal{X} \to \mathbb{R}^m$ that only deviates from the reference controller u_{ref} when the condition (5b) is violated. We will refer to (5b) as the *true certifying constraint* and (5) as the *oracle CF-QP*. When specifically using CBFs or CLFs in place of C, we may refer to (5) as the oracle CBF-QP and CLF-QP, respectively.

The oracle CF-QP requires perfect knowledge of the system dynamics since the Lie derivatives of C appear in the constraint. Instead, we can use the nominal model and replace $L_f C(x)$ and $L_g C(x)$ with $L_{\tilde{f}} C(x)$ and $L_{\tilde{g}} C(x)$ respectively, the Lie derivatives of C with respect to the nominal model. We call this a *nominal model-based CF-QP*.

The primary assumption we make in this paper is that we have access to the certificate function C that is valid for the true plant. This assumption ensures that a control policy exists to keep the true plant (1) in compliance with the system-critical constraint. However, even when a valid certificate function is available, obtaining such a control policy is not straightforward due to the lack of direct access to f and g in the true certifying constraint (5b). Due to the mismatch between the true plant dynamics and the nominal model, the nominal model-based CF-QP also does not provide any guarantee that the system-critical constraint will be met under the filtered control input. To examine this, the true certifying constraint in (5b) is expressed using the nominal model as follows:

$$\underbrace{L_{\tilde{f}} C(x) + L_{\tilde{g}} C(x) u}_{\widetilde{\dot{C}}(x,u)} + \Delta(x,u) + \gamma(C(x)) \geq 0, \tag{6}$$

where $\widetilde{\dot{C}}(x,u)$ is the Lie derivative of C based on the nominal model, and the *model uncertainty term* Δ [19], [26], is defined for each $x \in \mathcal{X}$, $u \in \mathbb{R}^m$ as

$$\begin{aligned}\Delta(x,u) :=& (L_f C - L_{\tilde{f}} C)(x) + (L_g C - L_{\tilde{g}} C)(x) u \\ =& \begin{bmatrix} L_{\Delta f} C(x) & L_{\Delta g} C(x) \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix}.\end{aligned} \tag{7}$$

Note that like the original constraint (5b), Δ is also affine in the control input u.

In the next section, we introduce a method developed in our prior work to estimate Δ from data collected from the true plant [7], [8]. Note that the data for Δ can be gathered from state trajectories without needing access to f and g. This can be achieved by evaluating $\dot{C}(x,u)$ along the trajectories using numerical differentiation and subtracting $\widetilde{\dot{C}}(x,u)$. By employing the estimate of Δ derived from the data, we can design a data-driven certifying filter that offers a high probability of satisfying (5b).

Remark 1. Discovering valid certificate functions for uncertain systems is far from trivial and is, in fact, an active area of research [47]–[52]. Our contribution runs parallel to this line of research, and in fact, our work complements these efforts, as only when the design of the certificate function and the design of the certifying filter are combined, can the certifying filter for uncertain systems be effectively implemented. In our work, we employ the nominal model to find CBFs and CLFs to be used as certificate functions. Thus, this procedure implicitly assumes that the nominal model is sufficiently accurate in its approximation of the true plant to enable the identification of a valid CBF or CLF. This assumption is also present in prior works that most closely align with our research [12], [13], [16], [25], [26]. This approach is considered reasonable for feedback linearizable systems with known relative degree, owing to the inherent robustness properties of CBFs and CLFs [53], [54]. Indeed, the practice of using first-principle nominal models for designing CBFs is widely adopted for numerous complex robotics systems [55]–[57].

IV. Data-driven Certifying Filters

The data-driven certifying filters we introduce in this section employ Gaussian Process (GP) regression to learn the estimate of Δ from data. We first provide a brief background on GP regression.

A. Gaussian Process Regression

A Gaussian Process is a random process where any finite collection of samples has a joint Gaussian distribution. The process is characterized by the mean function $q : \bar{\mathcal{X}} \to \mathbb{R}$ and the covariance (or kernel) function $k : \bar{\mathcal{X}} \times \bar{\mathcal{X}} \to \mathbb{R}$, where $\bar{\mathcal{X}}$ represents the input domain of the process.

GP regression is a Bayesian approach for regressing an unknown function $h : \bar{\mathcal{X}} \to \mathbb{R}$ by assuming that h is a sample from a GP, namely, $h \sim \mathcal{GP}(q,k)$. This implies that the prior distribution of $h(\bar{x}_*)$, where $\bar{x}_* \in \bar{\mathcal{X}}$ is an unseen query point, is given by $\mathcal{N}(q(\bar{x}_*), k(\bar{x}_*, \bar{x}_*))$. For our application, $\bar{\mathcal{X}} = \mathcal{X} \times \mathbb{R}^m$, where $\mathbb{R}^m$ represents the control input space, $\bar{x}_* = (x_*, u_*)$, and the unknown function we aim to regress is Δ defined in (7). We assume the mean function $q \equiv 0$ since the prior information, which is based on the nominal model, is already captured in the term $\widetilde{\dot{C}}(x,u)$ in (6).

With the dataset of noisy measurements of Δ, denoted by $\bar{z}_j$ at query $\bar{x}_j$, given as $\mathbb{D}_N := \{\bar{x}_j, \bar{z}_j\}_{j=1}^N = \{(x_j,u_j), \Delta(x_j,u_j) + \epsilon_j\}_{j=1}^N$, a prediction of Δ at $\bar{x}_*$ is derived from the joint distribution of $[\Delta(\bar{x}_1), \cdots, \Delta(\bar{x}_N), \Delta(\bar{x}_*)]^\top$ conditioned on the dataset

$\mathbb{D}_N$. Here, $\epsilon_j \sim \mathcal{N}(0, \sigma_n^2)$ is white measurement noise, with $\sigma_n > 0$. This conditional distribution at the query point $\bar{x}_*$ is called the *GP posterior*, whose mean and variance of the prediction of $\Delta(\bar{x}_*)$ are expressed as

$$\mu(\bar{x}_*|\mathbb{D}_N) = \mathbf{z}^\top (K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1} K_*^T, \tag{8}$$

$$\sigma^2(\bar{x}_*|\mathbb{D}_N) = k(x_*, x_*) - K_*(K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1} K_*^T, \tag{9}$$

where $K_{\mathbb{D}_N} \in \mathbb{R}^{N\times N}$ is the kernel matrix, whose $(i,j)^{th}$ element is $k(\bar{x}_i, \bar{x}_j)$, $K_* := [k(\bar{x}_*, \bar{x}_1), \cdots, k(\bar{x}_*, \bar{x}_N)] \in \mathbb{R}^N$, and $\mathbf{z} \in \mathbb{R}^N$ is the vector consisting of the dataset outputs, $\bar{z}_j$.

The kernel value, $k(\bar{x}, \bar{x}')$, quantifies the correlation between two query points $\bar{x}$ and $\bar{x}'$. A higher kernel value indicates a stronger correlation between the points, suggesting that the corresponding values of $\Delta(\bar{x})$ and $\Delta(\bar{x}')$ are more likely to be similar to each other. Thus, the choice of kernel k determines properties of the target function like its smoothness or Lipschitz constant [29]. For example, the square exponential kernel, which is among the most popular choices for kernels in the GP regression literature [5], attributes a higher correlation to points that are closer together in the input space, and the resulting samples of the GP are infinitely differentiable functions. The kernel can also capture prior structural knowledge of the target function [58]. In our case, we mainly want to exploit the fact that the target function Δ from (7) is control-affine. For this, we use the Affine Dot Product compound kernel presented in [8].

Definition 2. *Affine Dot Product Compound Kernel [8]:* Define $\mathbf{k} : \bar{\mathcal{X}} \times \bar{\mathcal{X}} \to \mathbb{R}$ given by

$$\mathbf{k}\big((x,u),(x',u')\big) := [1\ u^\top]\mathrm{Diag}(k_f(x,x'), k_{g_1}(x,x')\cdots, k_{g_m}(x,x'))\begin{bmatrix}1\\ u'\end{bmatrix}, \tag{10}$$

where $\mathrm{Diag}(\cdot)$ indicates the diagonal matrix whose diagonal terms consist of the entities in the paranthesis, as the *Affine Dot Product* (ADP) compound kernel of $(m+1)$ individual kernels $k_f, k_{g_1}, \cdots, k_{g_m} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

When $L_{\Delta f}C(x)$ and each element of $L_{\Delta g}C(x)$ in (7) is a sample from a GP defined by the individual kernels $k_f, k_{g_1} \cdots, k_{g_m}$, respectively, the model uncertainty term $\Delta(x,u)$ is a sample from a GP defined by the ADP kernel $\mathbf{k}$. Thus, using the ADP compound kernel, from (8) and (9), the GP posterior at a query point (x_*, u_*) is given as

$$\mu(x_*, u_*|\mathbb{D}_N) = \underbrace{\mathbf{z}^\top (K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1} K_{*U}^\top}_{=:\ \mathcal{M}(x_*|\mathbb{D}_N)} \begin{bmatrix}1\\ u_*\end{bmatrix}, \tag{11}$$

$$\sigma^2(x_*, u_*|\mathbb{D}_N) = \begin{bmatrix}1 & u_*^\top\end{bmatrix} \underbrace{\Big(K_{**} - K_{*U}(K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1} K_{*U}^T\Big)}_{=:\ \Sigma(x_*|\mathbb{D}_N)} \begin{bmatrix}1\\ u_*\end{bmatrix}, \tag{12}$$

where $K_{**} = \mathrm{Diag}\left(k_f(x_*,x_*), \cdots, k_{g_m}(x_*,x_*)\right) \in \mathbb{R}^{(m+1)\times(m+1)}$, and $K_{*U} \in \mathbb{R}^{(m+1)\times N}$ is given by

$$K_{*U} := \begin{bmatrix} k_f(x_*,x_1) & \cdots & k_f(x_*,x_N) \\ k_{g_1}(x_*,x_1) & \cdots & k_{g_1}(x_*,x_N) \\ \vdots & & \\ k_{g_m}(x_*,x_1) & \cdots & k_{g_m}(x_*,x_N) \end{bmatrix} \circ \begin{bmatrix}\mathbf{1}^{1\times N}\\ U_N\end{bmatrix}, \tag{13}$$

where $\circ$ indicates the element-wise product, and $U_N := [u_1\ \cdots\ u_N] \in \mathbb{R}^{m\times N}$. Note that $\Sigma(x_*|\mathbb{D}_N)$ is positive definite when the individual kernels $k_f, k_{g_1} \cdots, k_{g_m}$ are positive definite kernels and $\sigma_n > 0$ [8].

One of the most significant advantages of using GP regression is that it generates predictions of the target function value in the form of a probability distribution, rather than deterministically, based on (11) and (12). This allows for the computation of a probabilistic bound on the true value of $\Delta(x_*, u_*)$ using $\mu(x_*, u_*|\mathbb{D}_N)$ and $\sigma(x_*, u_*|\mathbb{D}_N)$:

Assumption 1. For a given $\delta \in (0,1)$, there exists a constant $\beta > 0$ such that

$$\mathbb{P}\Big\{\Big|\mu(x_*, u_*|\mathbb{D}_N) - \Delta(x_*, u_*)\Big| \le \beta\sigma(x_*, u_*|\mathbb{D}_N)\Big\} \ge 1-\delta, \tag{14}$$

for all $x_* \in \mathcal{X},\ u_* \in \mathbb{R}^m$.

Numerous existing works have conducted theoretical analyses to determine the conditions under which Assumption 1 holds and to identify the values of β. The term $\mu(x_*, u_*|\mathbb{D}_N) + \beta\sigma(x_*, u_*|\mathbb{D}_N)$ is referred to as the GP upper confidence bound (GP-UCB). It serves as an upper bound for $\Delta(x_*, u_*)$, with a high probability that the true value of $\Delta(x_*, u_*)$ is less than or equal to this bound [28]. Similarly, $\mu(x_*, u_*|\mathbb{D}_N) - \beta\sigma(x_*, u_*|\mathbb{D}_N)$ is the lower confidence bound of $\Delta(x_*, u_*)$. It is important to note that verifying the right value of β is not the primary focus of this work, and we direct interested readers to the relevant literature for further details [28], [29], [59]–[61].

B. Second-order Cone Program-based Certifying Filters

With the bound provided in (14), we can now present a data-driven certifying filter that offers a high probability guarantee of satisfying (5b) based on the learned GP model of Δ. By employing the lower bound of $\Delta(x,u)$, we construct a *certifying chance constraint* that can be evaluated without explicit knowledge of the true plant's dynamics:

$$L_{\tilde{f}}C(x) + L_{\tilde{g}}C(x)u + \mu(x,u|\mathbb{D}_N) - \beta\sigma(x,u|\mathbb{D}_N) + \gamma(C(x)) \ge 0. \tag{15}$$

If the constraint (15) is satisfied, from Assumption 1, we have a guarantee that the true certifying constraint in (5b) is satisfied with a probability of at least $1-\delta$.

Note that from the affine structure of the mean expression in (11), we get

$$\mu(x,u|\mathbb{D}_N) = \mathcal{M}(x|\mathbb{D}_N)\begin{bmatrix}1\\ u\end{bmatrix} = \begin{bmatrix}\widehat{L_{\Delta f}C}(x) & \widehat{L_{\Delta g}C}(x)\end{bmatrix}\begin{bmatrix}1\\ u\end{bmatrix},$$

where

$$\widehat{L_{\Delta f}C}(x) := \mathcal{M}(x|\mathbb{D}_N)_{[1]}, \quad \widehat{L_{\Delta g}C}(x) := \mathcal{M}(x|\mathbb{D}_N)_{[2:(m+1)]}.$$

We define

$$\begin{aligned}\widehat{L_fC}(x|\mathbb{D}_N) &:= L_{\tilde{f}}C(x) + \widehat{L_{\Delta f}C}(x) \in \mathbb{R},\\ \widehat{L_gC}(x|\mathbb{D}_N) &:= L_{\tilde{g}}C(x) + \widehat{L_{\Delta g}C}(x) \in \mathbb{R}^{1\times m}.\end{aligned} \tag{16}$$

Using these expressions, (15) can be represented as

$$\beta\sigma(x,u|\mathbb{D}_N) \leq \begin{bmatrix} \widehat{L_f C}(x|\mathbb{D}_N)+\gamma(C(x)) & \widehat{L_g C}(x|\mathbb{D}_N) \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix}. \tag{17}$$

From the quadratic structure of the variance expression in (12) and $\Sigma(x|\mathbb{D}_N)$ being positive definite, we can conclude that (17) is a second-order cone constraint.

This constraint is then incorporated into a chance-constrained reformulation of the CF-QP [7], [8]:

GP-CF-SOCP:

$$u^*(x) = \underset{u\in U}{\arg\min} \|u - u_{\text{ref}}(x)\|_2^2 \quad \text{s.t.} \tag{18}$$
$$L_{\tilde{f}} C(x) + L_{\tilde{g}} C(x)u + \mu(x,u|\mathbb{D}_N) - \beta\sigma(x,u|\mathbb{D}_N) + \gamma(C(x)) \geq 0,$$

wherein by leveraging the control-affine structure in the GP regression, we obtain a convex optimization problem, which is a second-order cone program (SOCP) that can be solved efficiently at high-frequency rates using modern solvers. For the full proof that (18) is an SOCP, readers are referred to [8]. When specifically using CBFs or CLFs in place of C, we refer to (18) as GP-CBF-SOCP and GP-CLF-SOCP, respectively.

The guarantee that the true certifying constraint will be satisfied with high probability exists only when the SOCP filter in (18) is feasible. However, this program can become infeasible for two reasons. Firstly, there might not be any u such that the prediction uncertainty in the left-hand side of (17) is adequately small compared to its right-hand side. This suggests that the dataset for the GP regression is insufficient to characterize the model uncertainty term Δ with high confidence. In this case, it may be necessary to collect more data to reduce the prediction uncertainty and ensure the feasibility of (17). A detailed analysis of the conditions under which (18) is feasible is conducted in [7]. Secondly, most robotic systems have bounded control input limits, either due to their physical actuation limits or safety concerns. This input bound, represented as $U \subset \mathbb{R}^m$, further constrains the feasible set of (18) and may render it infeasible.

During the deployment of the SOCP filter on real-world systems, it is often impossible to perfectly eliminate infeasibility. However, an effective strategy to address cases when infeasibility occurs is to use a backup control input computed by the following second-order cone program:

$$u^*(x) = \underset{u\in U}{\arg\min} \left(\beta\sigma(x,u|\mathbb{D}_N) - \widehat{L_g C}(x|\mathbb{D}_N)u\right). \tag{19}$$

This selects a control input within the input bound that minimizes the violation of the constraint (17).

The main challenge in executing the SOCP filter (18) online lies not in solving the optimization problem, but rather in the computationally demanding evaluation of σ when the size of the dataset is large. The time complexity of the matrix inverse in (12), $(K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1}$, is $\mathcal{O}(N^3)$, while the remaining matrix multiplication involved in evaluating $\Sigma(x_*|\mathbb{D}_N)$ has a time complexity of $\mathcal{O}(N^2)$. Although the matrix inversion can be performed offline, when the dataset is large, the $\mathcal{O}(N^2)$ complexity still remains challenging. This issue primarily motivates the development of Sparse GP literature [30] and the constraint-guided online data selection algorithm proposed in this paper, which can reduce the computational complexity to a linear dependence on N.

C. Running Example: 2D Polynomial System (Polysys)

We now introduce a simple running example system, referred to as Polysys, which is utilized throughout the paper. It is important to note that this low-dimensional toy example is not intended to showcase the computational advantage of our method, a topic we will present in Section VI. Instead, its purpose is to provide a walk-through of the inner workings of our approach for the readers. To achieve this, we have access to the true plant dynamics, allowing us to compare our method with the ideal oracle certifying filter. Moreover, the dataset constructed for the data-driven certifying filters is not meant to represent a realistic dataset. Instead, it is a simplistic dataset designed for easy comprehension by the readers.

The dynamics of the system, whose vector fields are polynomial functions of the state x, are given by:

$$\dot{x} = \begin{bmatrix} f_1^T v \\ f_2^T v \end{bmatrix} + \begin{bmatrix} 1+g_{11}^T v & g_{12}^T v \\ g_{21}^T v & 1+g_{22}^T v \end{bmatrix} u, \tag{20}$$

where $x = [x_1\ x_2]^\top$ is the state, $u = [u_1\ u_2]^\top$ is the control input, $v = [x_1\ x_2\ x_1^2\ x_1x_2\ x_2^2\ x_1^3\ x_1^2x_2\ x_1x_2^2\ x_2^3] \in \mathbb{R}^9$ is a vector that aggregates the monomials of the state, and each of $f_1, f_2, \cdots, g_{22}$ are randomly generated coefficient vectors in $\mathbb{R}^9$. We introduce model uncertainty to the true plant by perturbing the coefficient vectors from the nominal model.

In this example, we aim to design a control policy that stabilizes the system to the zero equilibrium point. To achieve this by using the certifying filter, we design the CLF $V(x) = x^\top P x$ as the certificate function, where P is the solution of the Algebraic Riccati Equation for the linearized system of the nominal model (20) around $x = 0$. In this example, we set $u_{\text{ref}}(x) \equiv 0$ since we do not have any other explicit tasks to achieve. As shown in Figure 1, we can see that the oracle CLF-QP (orange) is able to stabilize the state to the equilibrium, confirming that the CLF is a valid certificate function for the true plant. However, due to the model uncertainty we introduce to the true plant, the nominal model-based CLF-QP (green) fails to stabilize the system.

We next show the application of the GP-CLF-SOCP in (18) on this example. We first construct the dataset $\mathbb{D}_N$ in order to apply GP regression to Δ. We partition the subspace of the state space $[-2,2] \times [-2,2]$ into the coarse state grid of size $(10,10)$. At every vertex x_j of the state grid, we apply the randomly sampled control input u_j to simulate the system (20) for a sampling time Δt and collect a single data point $(\bar{x}_j = (x_j, u_j), \bar{z}_j)$. We account for the numerical differentiation error in obtaining $\bar{z}_j$ as measurement noise. In addition to the data from the coarse grid, we also incorporate some densely populated data points centered at a few selected state and action pairs, (x_a, u_a). Around each of these points, a dense state-control grid is created by gridding up $[x_{a,1} - \delta, x_{a,1} + \delta] \times [x_{a,2} - \delta, x_{a,2} + \delta] \times [u_{a,1} - \delta, u_{a,1} + \delta] \times [u_{a,2} - \delta, u_{a,2} + \delta]$ in (2, 2, 2, 2) grid, where we set $\delta = 0.1$. This results in a total of 81 data points collected at each (x_a, u_a). In the subsequent

sections describing the Polysys example, we refer to the data points generated from a single dense grid as *a data cluster*. Combined together, we get in total $N = 361$ data points, visualized by their projection to the state space in Figure 1.

We use GP regression to fit Δ from the dataset presented above, using the ADP compound kernel with isotropic squared exponential kernels as components. Then, we apply the GP-CLF-SOCP of (18) to control the system. As shown in Figure 1, the GP-CLF-SOCP using the full dataset (black dashed line) is able to stabilize the system to near the origin despite the uncertainty in the true plant dynamics.

V. Constraint Guided Online Data Selection

In this section, we present the core contribution of our paper: a constraint-guided online data selection algorithm that improves the time complexity of GP inference for the GP-CF-SOCP from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

Using the entire dataset to evaluate $\sigma^2(x_*, u_*|\mathbb{D}_N)$ would yield minimal uncertainty for any query point x_*, u_*, as we would utilize all available information, but this comes at the cost of high computational demand. One way to mitigate this computational burden involves constructing an offline model that approximates the precise Gaussian Process (GP) inference, with the goal of making accurate predictions for any new query points encountered during runtime. However, our approach, similar to many existing Sparse GP methods, is based on the idea that it is not necessary to reduce the uncertainty globally [62]–[64]. Instead, we aim to reduce the uncertainty for specific input classes relevant to our problem. The former approach, known as *induction*, aims to regress the function with high quality across the entire input space. In contrast, our approach, which is called *transduction*, focuses on learning only for specific test points that we care about [31]. Revisiting the learning objective in our problem, we seek to find $u^*(x)$ such that the certifying chance constraint (15) is feasible. Therefore, our data selection algorithm is designed to efficiently achieve this goal.

To facilitate the presentation of our algorithm, we first introduce some simplified notations and preliminaries that will be used in this section. We also present a sufficient condition for the feasibility of GP-CF-SOCP, from which we derive the main control input direction we want to characterize. This control input direction is the foundation upon which we apply the concept of transduction in our data selection algorithm.

A. Preliminaries

1) Simplified notations for kernels: We use

$$\mathbf{k}_{ij} := \mathbf{k}\left((x_i, u_i), (x_j, u_j)\right),$$
$$\mathbf{k}_{**}(x, u) := \mathbf{k}\left((x, u), (x, u)\right),$$
$$\mathbf{k}_{*i}(x, u) := \mathbf{k}\left((x, u), (x_i, u_i)\right),$$

and $\mathbf{k}_i := \mathbf{k}_{ii}$ as simplified notations, where (x_i, u_i) is an input point in $\mathbb{D}_N$. We also consider the compound kernel that captures only the control vector field-relevant part:

$$\mathbf{k}^u((x, u), (x', u')) := u^\top \mathrm{Diag}(k_{g_1}(x, x'), \cdots, k_{g_m}(x, x'))u'. \tag{21}$$

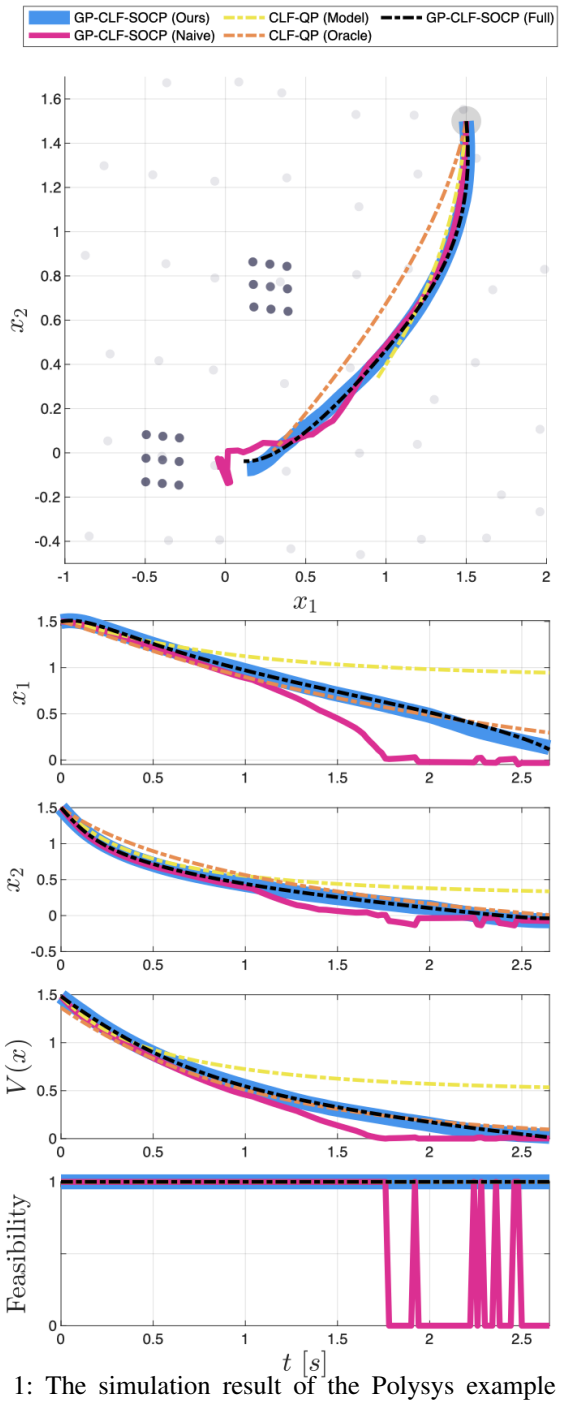

Fig. 1: The simulation result of the Polysys example under various controllers: the nominal model-based CLF-QP (yellow), the oracle CLF-QP (orange), the GP-CLF-SOCP using full data (black), the GP-CLF-SOCP using naive data selection (magenta) discussed in Sec. V-C, the GP-CLF-SOCP using our main data selection algorithm (blue) discussed in Sec. V-D, both using the same number of online data, $M = 40$. The topmost plot illustrates the trajectory's progression in the state space for 2.6 seconds, with an initial state of $x_0 = [1.5\ 1.5]^\top$, while the data is depicted as grey dots. The four subplots on the bottom show the state x_1, x_2, the CLF values, and the feasibility of the QP and SOCP in time, respectively. While the naive approach often faces infeasibility and fails to stabilize the system close to the origin, our approach effectively selects an online dataset that secures the feasibility of the SOCP.

Note that $\mathbf{k}((x,u),(x',u')) = k_f(x,x') + \mathbf{k}^u((x,u),(x',u'))$ from Definition 2. Similarly, we define

$$\mathbf{k}^u_{**}(x,u) := \mathbf{k}^u((x,u),(x,u)),$$
$$\mathbf{k}^u_{*i}(x,u) := \mathbf{k}^u((x,u),(x_i,u_i)).$$

2) Alternative expression for the GP posterior variance (12): Using simplified notations, we can express

$$\begin{bmatrix}1 & u_*^\top\end{bmatrix} K_{*U} = [\mathbf{k}_{*1}(x_*,u_*) \ \cdots \ \mathbf{k}_{*N}(x_*,u_*)],$$

and (12) becomes

$$\sigma^2(x_*,u_*|\mathbb{D}_N) \tag{22}$$
$$= \mathbf{k}_{**}(x_*,u_*) - [\mathbf{k}_{*1} \ \cdots \ \mathbf{k}_{*N}](K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1}\begin{bmatrix}\mathbf{k}_{*1}\\ \vdots \\ \mathbf{k}_{*N},\end{bmatrix}$$

with (x_*,u_*) dropped in $\mathbf{k}_{*i}$ for simplicity. Note that the first term on the right-hand side is contributed from the GP prior, and the choice of the data only affects the second term.

3) Sufficient Condition for Pointwise Feasibility of GP-CF-SOCP: The expression of the certifying chance constraint in (17) highlights the tradeoff required to evaluate its feasibility, which lies between the prediction uncertainty of the GP regression on the left-hand side and the mean-estimate of the true certifying constraint on the right-hand side. This structure is useful for verifying the following sufficient condition for the feasibility of (17).

Lemma 1. Given a dataset $\mathbb{D}_N$, for a point $x \in \mathcal{X}$, If there exists a constant $\alpha > 0$ such that the following inequality holds,

$$\beta\,\sigma\Big(x, \alpha\widehat{L_gC}(x|\mathbb{D}_N)^\top \big|\mathbb{D}_N\Big) < \alpha\left\|\widehat{L_gC}(x|\mathbb{D}_N)\right\|^2 \tag{23}$$

then the GP-CF-SOCP in (18) is feasible. The feasible control input can be found by taking $u = \alpha'\widehat{L_gC}(x|\mathbb{D}_N)^\top$ with sufficiently large $\alpha' > 0$.

Proof. See Appendix A. □

The main implication of the above lemma is that the feasibility of (18) can be assessed by examining the size of the prediction uncertainty, σ, in just one control input direction, specifically the direction of the mean-based estimate of $L_gC(x)$, denoted as $\widehat{L_gC}(x|\mathbb{D}_N)$. This direction is particularly important because according to what the mean prediction of the GP tells, it is the control input direction in which we can most effectively regulate the value of $C(x)$. If the prediction uncertainty is sufficiently small in this direction, by taking the control input in this direction with large enough magnitude, we can ensure (18) to be feasible.

B. Data Selection Objective

We seek to design an online data selection algorithm, that selects a subset of data from the entire dataset, $\mathbb{D}_M(x) \subset \mathbb{D}_N$, at every sampling time at the current state x. Once M online data points are determined, the GP-CF-SOCP in (18) is solved with the online dataset $\mathbb{D}_M(x)$ in place of $\mathbb{D}_N$, to determine the filtered control input $u^*(x)$ which will be applied to the system next. Among the data points in the full dataset, we want to select a limited number of points that are most helpful in characterizing the control direction that secures the feasibility of the certifying chance constraint in (15).

We attempt to achieve this by utilizing the result of Lemma 1, trying to make sure that condition (23) is met with the limited M data points we are allowed to use. Adopting the approach of transduction, the goal of the data selection is to reduce the uncertainty in the direction of $\widehat{L_gC}(x|\mathbb{D}_M)^\top$, i.e., select $\mathbb{D}_M(x)$ which best reduces $\sigma\Big(x, \alpha\widehat{L_gC}(x|\mathbb{D}_M)^\top|\mathbb{D}_M\Big)$ for sufficiently large α. However, we do not know how sufficiently large α needs to be to render (23) feasible prior to selecting $\mathbb{D}_M(x)$ and actually solving the SOCP. Therefore, we eliminate the dependency on the magnitude of α by considering the following problem:

$$\arg\min_{\mathbb{D}_M(x)} \lim_{\alpha\to\infty} \frac{1}{\alpha}\sigma\Big(x, \alpha\widehat{L_gC}(x|\mathbb{D}_M)^\top|\mathbb{D}_M\Big). \tag{24}$$

Note that we drop the dependency of $\mathbb{D}_M$ on x whenever it is obvious, for notational simplicity. From the expression of the variance in (22), we can derive the following lemma that transforms the objective function above into a form without the appearance of α:

Lemma 2. The optimization problem (24) can be equivalently expressed as

$$\arg\max_{\mathbb{D}_M\subset\mathbb{D}_N} F_{\mathbb{D}_M}(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top), \tag{25}$$

where $F_{\mathbb{D}_M}(x,u) :=$

$$[\mathbf{k}^u_{*1}(x,u) \ \cdots \ \mathbf{k}^u_{*M}(x,u)](K_{\mathbb{D}_M} + \sigma_n^2 I)^{-1}\begin{bmatrix}\mathbf{k}^u_{*1}(x,u)\\ \vdots \\ \mathbf{k}^u_{*M}(x,u)\end{bmatrix}, \tag{26}$$

which is the second order term in the control input u of the posterior variance $\sigma^2(x,u|\mathbb{D}_M)$.

Proof. See Appendix B. □

Thus, we will consider $F_{\mathbb{D}_M}(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top)$ as the *objective function of the data selection algorithm.*

Remark 2. Since we do not have access to $\widehat{L_gC}(x|\mathbb{D}_M)$ prior to determining $\mathbb{D}_M$, we can replace $\widehat{L_gC}(x|\mathbb{D}_M)$ in $F_{\mathbb{D}_M}$ with $\widehat{L_gC}(x|\mathbb{D}_N)$, where $\mathbb{D}_N$ is the entire dataset. Note that $\widehat{L_gC}(x|\mathbb{D}_N)$ only requires the computation of $\mu(x,u|\mathbb{D}_N)$ but not $\sigma(x,u|\mathbb{D}_N)$. Since $\mathbf{z}^\top(K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1}$ in (11) can be precomputed offline, the time complexity of evaluating $\widehat{L_gC}(x|\mathbb{D}_N)$ online is $\mathcal{O}(N)$. When N is very large, it may be impractical or computationally infeasible to evaluate $\mathbf{z}^\top(K_{\mathbb{D}_N} + \sigma_n^2 I)^{-1}$ offline since it requires us to compute the inverse of the matrix. In such cases, an effective approximation for $\widehat{L_gC}(x|\mathbb{D}_M)$ can still be achieved by using $\widehat{L_gC}(x|\mathbb{D}'_M)$, where $\mathbb{D}'_M$ represents the dataset selected online at the previous time step.

C. Naive approach

Before we proceed to present the main algorithm of the paper, let's take a moment to build a better understanding of

the data points we wish to include in $\mathbb{D}_M(x)$. To facilitate this discussion and simplify our thought process, consider a scenario where all data points in $\mathbb{D}_N$ are not correlated with one another, meaning that $\mathbf{k}_{ij} = \mathbf{k}((x_i,u_i),(x_j,u_j)) = 0$ for all $i \neq j$. Additionally, let's assume there is no noise in the data, so $\sigma_n = 0$. In this simplified case, $K_{\mathbb{D}_N} + \sigma_n^2 I = \mathrm{Diag}(\mathbf{k}_1, \cdots, \mathbf{k}_N)$, and from (26) it holds that

$$
\begin{aligned}
&F_{\mathbb{D}_M}(x,u) \\
&= [\mathbf{k}^u_{*1}(x,u) \ \cdots \ \mathbf{k}^u_{*M}(x,u)]\mathrm{Diag}\left(\frac{1}{\mathbf{k}_1},\cdots,\frac{1}{\mathbf{k}_M}\right)\begin{bmatrix}\mathbf{k}^u_{*1}(x,u)\\ \vdots \\ \mathbf{k}^u_{*M}(x,u)\end{bmatrix} \\
&= \begin{bmatrix}\frac{\mathbf{k}^u_{*1}(x,u)}{\sqrt{\mathbf{k}_1}} & \cdots & \frac{\mathbf{k}^u_{*M}(x,u)}{\sqrt{\mathbf{k}_M}}\end{bmatrix}\begin{bmatrix}\frac{\mathbf{k}^u_{*1}(x,u)}{\sqrt{\mathbf{k}_1}}\\ \vdots \\ \frac{\mathbf{k}^u_{*1}(x,u)}{\sqrt{\mathbf{k}_1}}\end{bmatrix} = \sum_{i=1}^{M}\left(\frac{\mathbf{k}^u_{*i}(x,u)}{\sqrt{\mathbf{k}_i}}\right)^2.
\end{aligned}
$$

We define a kernel-based alignment measure as

$$
n_i(x,u) := \frac{|\mathbf{k}^u_{*i}(x,u)|}{\sqrt{\mathbf{k}_i}} = \frac{|\mathbf{k}^u((x,u),(x_i,u_i))|}{\sqrt{\mathbf{k}((x_i,u_i),(x_i,u_i))}}, \tag{27}
$$

which results in

$$
F_{\mathbb{D}_M}\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right) = \sum_{i=1}^{M} n_i^2\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right). \tag{28}
$$

Therefore, we can optimize $F_{\mathbb{D}_M}\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ simply by selecting M points from $\mathbb{D}_N$ that exhibit maximum values of $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$. The time complexity of finding such points is $\mathcal{O}(N)$, which can be achieved using efficient algorithms, such as a quick selection.

Equation (28) highlights that the alignment measure $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ is the measure of the relevance of the data point (x_i,u_i) to the feasible direction of the certifying chance constraint. Here, we offer a concise explanation of the geometric interpretation of this measure.

For kernels used in GP regression, note that the kernel value of two inputs, $k(x,x')$ can be interpreted as an inner product between the feature vectors of x and x', i.e. $k(x,x') = \varphi(x)\cdot\varphi(x')$ [5]. For the ADP kernel in Definition 2, denoting the feature vectors for individual kernels as φ_f, $\varphi_{g_1},\cdots,\varphi_{g_m}$, we can express the ADP kernel's feature vector as $\boldsymbol{\varphi}(x,u) := [\varphi_f(x)\ \varphi_{g_1}(x)\ \cdots\ \varphi_{g_m}(x)]\begin{bmatrix}1\\ u\end{bmatrix}$. Consequently, we get

$$
n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right) = \lim_{\alpha\to\infty}\frac{\left|\boldsymbol{\varphi}(x, \alpha\widehat{L_gC}(x|\mathbb{D}_M)^\top)\cdot\boldsymbol{\varphi}(x_i,u_i)\right|}{\alpha\sqrt{\boldsymbol{\varphi}(x_i,u_i)\cdot\boldsymbol{\varphi}(x_i,u_i)}},
$$

from (27), where we get rid of the autonomous vector field relevant part from the numerator in (27) by taking the limit of $\alpha\to\infty$. Thus, $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ captures how well the data point is aligned in the feature space of the ADP kernel with the feasible input direction.

In summary, the naive approach, which selects M points with maximum values of $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ from the dataset $\mathbb{D}_N$, optimally achieves the objective in (24) under the ideal conditions of an uncorrelated dataset and absence of measurement noise. However, these assumptions do not accurately represent the characteristics of real-world datasets. In practice, data from actual systems often have a high correlation because sampled data points from trajectories are sequential and share similar properties due to their close proximity in time and space.

We use the Polysys example to highlight the failure of the naive approach in handling datasets that deviate from ideal conditions, particularly those containing self-correlated data points. Our demonstration reveals that the naive approach may choose an unsuitable $\mathbb{D}_M$, rendering the SOCP filter infeasible. This limitation motivates the development of a more advanced data selection algorithm, which we present in the next subsection.

Running Example–Polysys (Cont'd): As described in Section IV-C, the dataset created for the Polysys example contains highly correlated data points, especially in the data clusters. Figure 2 (a) illustrates a failure case of the naive algorithm. In the first row of Figure 2 (a), we visualize the selected data points $\mathbb{D}_M(x)$ at a query state x under various values of M. The second row represents the prediction uncertainty $\beta\sigma(x,u)$ in control-input space as an ellipse, and $\widehat{L_gC}(x|\mathbb{D}_M)$ as a dashed magenta line, thereby illustrating the competitive relationship between the left-hand side (ellipse) and the right-hand side (magenta line) of the certifying chance constraint (17).

Since the naive approach greedily selects the points that maximize $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ without considering the correlation between them, the selected data points are sourced from the data cluster that is close to the query state. The effect of using such highly self-correlated data points as $\mathbb{D}_M$ is shown in the second row of the figure. It demonstrates that even after increasing the size of M from 40 to 60, the uncertainty ellipse barely reduces its size, leading to the infeasibility of the SOCP. Clearly, selecting such concentrated data points does not provide additional information, which intuitively illustrates why the naive approach can fail.

D. Main Algorithm

Selecting the data points in the dataset $\mathbb{D}_N$ that maximize our objective function $F_{\mathbb{D}_M}(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top)$ when the dataset is self-correlated is in fact a combinatorial optimization problem which is NP-hard [65]. This complexity occurs from the need, as seen in (26), to find the optimal subset of data that maximizes the correlation with the target point ($\mathbf{k}_{*j}$ in (26)), while minimizing the self-correlation within the subset (captured by $K_{\mathbb{D}_M}$). Therefore, directly optimizing for the objective function online is intractable. The result presented next, which is the main assertion of our paper, allows us to indirectly find a good candidate $\mathbb{D}_M$ by maximizing a lower bound of the objective function.

Theorem 1. For a given dataset $\mathbb{D}_M$ with $M \geq 2$, assume that there exists a constant $\epsilon \in [0,1)$ that satisfies

$$
\mathbf{k}_{ij}^2 < \epsilon^2 \mathbf{k}_i \mathbf{k}_j, \tag{29}
$$

for all $i,j = 1,\cdots,M$ and $i \neq j$, and

$$
\sigma_n^2 \leq \frac{\epsilon^2(M-1)\min_i \mathbf{k}_i}{1-\epsilon}. \tag{30}
$$

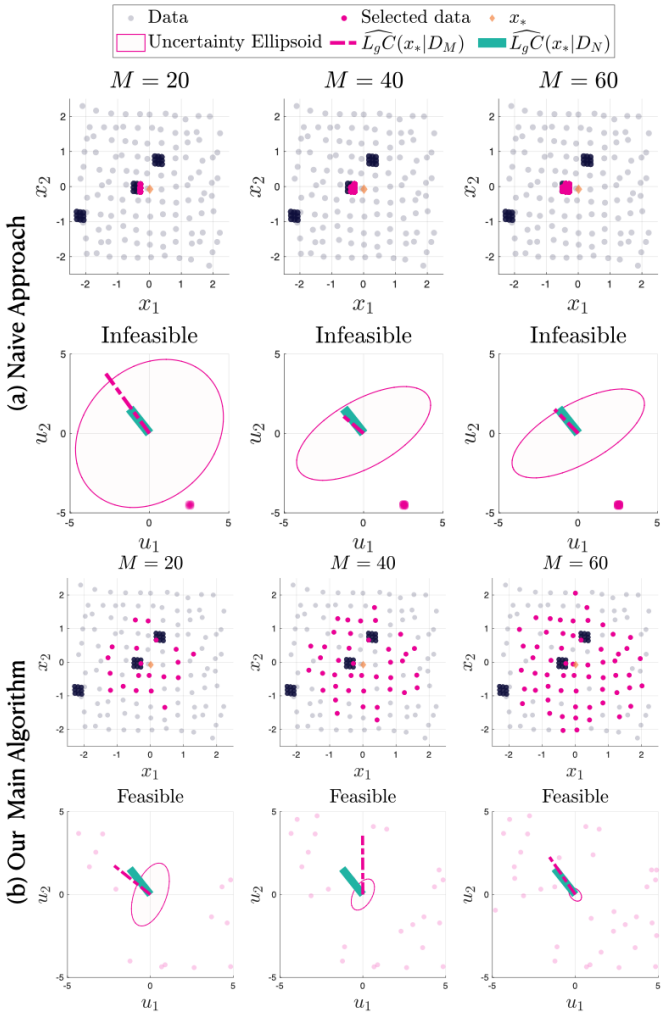

Fig. 2: Comparison between the two data selection strategies– (a) naive approach described in Section V-C and (b) our main algorithm described in Section V-D, on the Polysys running example system, with a varying number of online selected data points ($M = 20, 40, 60$). In each case, the first row visualizes the entire dataset $\mathbb{D}_N$ (grey dots) projected on the state space and the data points selected online $\mathbb{D}_M$ (magenta dots) according to the data selection algorithm at the query state $x = [0.011 \ \ -0.0756]^\top$ (orange diamond). The second row visualizes the selected points projected on the control input space (magenta dots), and the prediction uncertainty $\beta\sigma(x,u)$'s growth in the control space as an ellipse. We also visualize $\widehat{L_gC}(x|\mathbb{D}_N)$ and $\widehat{L_gC}(x|\mathbb{D}_M)$ as the dashed green and magenta lines, respectively. The ellipse and magenta line represent the growth of the right-hand side and left-hand side of (17), respectively. The feasibility of the chance certifying constraint can be deduced by evaluating the relative ratio of the magenta line's length to the ellipse's radial distance in the magenta line's direction. A smaller ratio suggests that a larger control input in the $\widehat{L_gC}(x|\mathbb{D}_M)$ direction is required to satisfy the chance constraint.

Then, $F_{\mathbb{D}_M}(x,u)$ is lower bounded by the inequality below

$$F_{\mathbb{D}_M}(x,u) \geq \frac{1-\epsilon}{1+\epsilon(M-2)} \sum_{i=1}^{M} n_i^2(x,u). \tag{31}$$

Note that the equality is satisfied when $\epsilon = 0$.

Proof. See Appendix C. □

The condition (29) requires the dataset to exhibit no more than a weak correlation, while condition (30) necessitates that the noise variance remains comparatively small with respect to the correlation threshold ϵ. It is worth noting that the latter condition becomes less stringent as the value of M increases. Under these conditions, Theorem 1 concludes that the lower bound of the objective function can be maximized by, again, selecting M points with maximum values of $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$.

Leveraging the result of Theorem 1, we aim to maximize the lower bound as a proxy for the original objective function, thereby rendering the problem more tractable. The essence of our main algorithm is to condition the dataset to satisfy the assumption in (29), ensuring that Theorem 1 holds, and then identify the data points for which $\sum_{i=1}^{M} n_i^2(x,u)$ is maximized.

We achieve this through a two-fold algorithm. First, during the offline phase, we compute a ready-to-use binary matrix $B \in \mathbb{R}^{N\times N}$, with elements defined as follows:

$$B_{ij} := \begin{cases} 1 & \text{if } \mathbf{k}_{ij}^2 < \epsilon^2 \mathbf{k}_i \mathbf{k}_j \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

The matrix B can be efficiently constructed by applying an ϵ-threshold to the matrix

$$\text{Diag}\left(\frac{1}{\sqrt{\mathbf{k}_1}}, \cdots, \frac{1}{\sqrt{\mathbf{k}_N}}\right)^\top K_{\mathbb{D}_N} \text{Diag}\left(\frac{1}{\sqrt{\mathbf{k}_1}}, \cdots, \frac{1}{\sqrt{\mathbf{k}_N}}\right) \tag{33}$$

This operation has a time complexity of $\mathcal{O}(N^2)$ but occurs during the offline stage, so it does not impact the online time complexity.

Next, in the online phase described in Algorithm 1, we first initialize a candidate dataset as the entire dataset (Line 4). We then sequentially add to the online dataset $\mathbb{D}_M$ the data point that has the maximum value of $n_i\left(x, \widehat{L_gC}(x|\mathbb{D}_M)^\top\right)$ among those in the candidate dataset (Line 6-7). As we select each point, we remove from the candidate dataset the points that have a correlation greater than ϵ relative to the selected point, by directly referring to the matrix B (Line 8).

Algorithm 1 has a time complexity of $\mathcal{O}(MN)$, as each operation in Line 6 and Line 8 inside the for loop is $\mathcal{O}(N)$. At each time step, after obtaining $\mathbb{D}_M$ from the proposed algorithm, we use this online dataset for the GP-CF-SOCP filter in (18) instead of using the entire dataset. This requires evaluating the matrix inverse in (11) and (12) online, which has a time complexity of $\mathcal{O}(M^3)$. Thus, with our proposed approach, obtaining the optimal filtered control input $u^*(x)$ has a total time complexity of $\mathcal{O}(NM + M^3)$, in terms of N and M. We are neglecting the time complexity of solving the SOCP since it does not depend on the number of data

Algorithm 1: Online Data Selection for GP-CF-SOCP

Input: Current state x, entire dataset $\mathbb{D}_N$, B defined in (32)
Output: Online dataset $\mathbb{D}_M$
$\mathbb{D}_M \leftarrow \emptyset$
$\mathcal{I}_{\text{candidate}} \leftarrow \{1, 2, \cdots, N\}$
for $k = 0;\ k < M;\ k = k + 1$ **do**
  $i_* \leftarrow \arg\max_{i \in \mathcal{I}_{\text{candidate}}} \ n_i(x, \widehat{L_g C}(x|\mathbb{D}_M)^\top)$
  $\mathbb{D}_M \leftarrow \mathbb{D}_M \cup (x_{i_*}, u_{i_*}, z_{i_*})$
  $\mathcal{I}_{\text{candidate}} \leftarrow \mathcal{I}_{\text{candidate}} \setminus \{j \in \mathcal{I}_{\text{candidate}} \mid B_{i_* j} == 0\}$
end

points. Given that we choose $M << N$ in practice, the time complexity of the GP-CF-SOCP safety filter combined with our online data selection algorithm is linear in N.

Remark 3. Choosing the value of the correlation threshold ϵ allows users to strike a balance between the *contribution* of the term $\sum_{i=1}^{M} n_i^2(x, u)$ and the *adverse impact* of self-correlation on the objective function $F_{\mathbb{D}_M}(x, u)$. With a value of $\epsilon = 1$, the data selection is identical to the naive approach. However, the right-hand side of (31) being zero indicates that the information gained from the selected data points can be significantly compromised by their self-correlations, potentially resulting in no contribution to the objective function at all. Conversely, $\epsilon = 0$ prohibits users from using data points with even the slightest correlation, which is impractical. Ideally, we should find the optimal value of ϵ that offers the best trade-off. However, determining the optimal ϵ is an NP-hard problem, as it shares the same problem complexity as maximizing the data selection objective $F_{\mathbb{D}_M}$ directly. A practical and effective strategy is to leverage prior knowledge of the full dataset to identify an acceptable ϵ value, for instance, by evaluating the histogram of $\frac{\mathbf{k}_{ij}^2}{\mathbf{k}_i \mathbf{k}_j}$ for the dataset and selecting an ϵ that corresponds to a reasonable quantile of data satisfying (29).

Running Example–Polysys (Cont'd): We investigate how Algorithm 1 selects data online and improves the downstream objective of enhancing the feasibility of the GP-CLF-SOCP through its self-correlation remedy in the Polysys example. We use $\epsilon = 0.95$ in the example, which is the minimum correlation between data points within a data cluster. Using this value prevents our main algorithm from selecting more than one point per data cluster. The first row of Figure 2 (b) displays that our main algorithm selects at most one data from each data cluster even as M increases. This correlation-aware behavior resulting from upper-bounding the maximum self-correlation of the selected data points induces the algorithm to select diverse data. Consequently, the prediction uncertainty, illustrated as the ellipse in the second row of the image, is reduced as M increases in all directions of u but, more importantly, it is primarily reduced in the direction of $\widehat{L_g C}(x|\mathbb{D}_N)$. Moreover, in the case of $M = 20$, it is notable that the algorithm prioritizes selecting data points whose control input values are well aligned in the direction of $\widehat{L_g C}(x|\mathbb{D}_M)$. As a result, the GP-CLF-SOCP controller utilizing the online dataset constructed by our main algorithm is feasible for all M in Figure 2.

E. Related Data Selection Methods

The point at issue of this paper is very related to the information-theoretic data subset selection [66], [67] and sensor placement [65], [68] problems, which are known to be NP-hard for many different objective functions, such as mutual information and conditional entropy [32], [69]. While our focus is on optimizing a particular certification-oriented measure (26) that differs from the information-theoretic objective functions, our optimization problem still suffers from the same combinatorial challenges, and solving (25) to optimality would be intractable for large datasets.

A reasonable alternative to our approach would be to form the online dataset $\mathbb{D}_M$ by greedily selecting, one at a time, the data points that maximize (25). This idea was applied to the sensor placement in [32] and has been used for data-driven control in [6], [70]. To approximately solve (25), this greedy selection method can be implemented with an asymptotic time complexity of $\mathcal{O}(NM^3)$. While this asymptotic complexity is only slightly worse than the $\mathcal{O}(NM)$ complexity of Algorithm 1, in practice we observe that the greedy method is too slow to perform the data selection online, even when using the locality and lazy evaluation speedups proposed in [65].

Another simpler approach would be to choose the k-nearest neighbors (k-NN) at each query state-action pair. However, given the control-affine structure of the target function Δ, it is not immediately clear which distance metric should be used for the k-NN to capture the most relevant information. The authors in [71] propose to use the *kernel distance* [72], [73], which is the Euclidean distance in the kernel feature space. Although simple, these k-NN selection approaches suffer from similar problems as our naive selection algorithm, as they do not consider the self-correlation of the dataset.

VI. Results

In this section, we apply our method to three specific examples, consisting of two numerical simulations and one hardware experiment. We refer to the GP-CF-SOCP filter using the full dataset as ***GP-CF-SOCP (Full)***, the GP-CF-SOCP using the online data constructed by the naive approach in Section V-C as ***GP-CF-SOCP (Naive)***, and the GP-CF-SOCP using the online data constructed by our main data selection algorithm as ***GP-CF-SOCP (Ours)***.

A. Running Example: Polynomial System (Cont'd)

The simulation results of Polysys under the GP-CLF-SOCP (Naive) and GP-CLF-SOCP (Ours) are evaluated in this study, extending the analysis in Section IV-C. To ensure a fair comparison, both controllers select 40 (M) data points from the full dataset, which has a total of 361 (N) data points, as constructed in Section IV-C. The results are presented in Figure 1, which shows that GP-CLF-SOCP (Ours) is feasible throughout the simulation period, imposing the probabilistic guarantee of stability to the closed-loop system. In contrast, the GP-CLF-SOCP (Naive) fails to do so, and the SOCP is infeasible very frequently with this approach. Note that when the SOCP is infeasible, the backup controller in (19)

is deployed, and the stability property that the certifying constraint is trying to impose is not guaranteed anymore.

Further analysis of the trajectory of the two controllers in the topmost plot of Figure 1 reveals two important behaviors. First, the GP-CLF-SOCP (Ours) (blue) exhibits a similar trajectory to that of GP-CLF-SOCP (Full) (black). This implies that the effect of the information loss due to using only the online-selected data points is negligible when employing our algorithm. Second, the GP-CLF-SOCP (Naive) controller (magenta) exhibits a more aggressive trajectory compared to GP-CLF-SOCP (Ours), as evidenced by the rapid decay of $V(x)$ and a swift change in state history starting at around $t = 1$s. This is when the naive algorithm begins selecting most of the data points in the densely populated data cluster near the origin. These observations demonstrate that the naive data selection leads to a large prediction uncertainty in the direction of $\widehat{L_g C}(x|\mathbb{D}_N)$, resulting in a considerably more conservative control policy than necessary. This also makes the controller more susceptible to infeasibility.

Note that the Polysys example is devised to provide a detailed walk-through of our method; thus, we do not benchmark the computation time of each method in this example. Given the relatively small number of data points used in this example, the computational efficiency gained from our method would not be easily noticeable.

B. High-dimensional System in Simulation: Five-link Walker

We explore the performance of our algorithm in a high-dimensional system, RABBIT [74], a planar five-link bipedal robot consisting of ten state variables. We demonstrate the effectiveness of our algorithm in achieving stable walking. The significance of our algorithm in reducing the computational demands of executing the certifying filter is highlighted.

RABBIT is a testbed system developed to study bipedal robot locomotion [74]. As depicted in Figure 3 (a), its configuration is represented by the generalized coordinate vector $q = [q_1, q_2, q_3, q_4, q_5]^\top$ consisting of the robot's joint angle variables. We adopt the mathematical model for RABBIT locomotion in [74] to design the simulation model of this system, where the state is defined as $x = [q, \dot{q}] \in \mathbb{R}^{10}$, and the control input is defined as $u \in \mathbb{R}^4$, consisting of the hip and knee motor torques for both legs. The torque saturation is set at $150Nm$. The hybrid system description of the robot's walking process consists of a single-support swing phase under a Lagrangian dynamics and a reset map defined by the rigid impact model, which switches the robot's state to the post-impact state upon the swing foot's impact with the ground.

The objective of the certifying filter is to achieve an exponentially stabilizing periodic gait for RABBIT, despite the effect of the impacts. To accomplish this, we employ a Rapidly Exponentially Stabilizing Control Lyapunov Function (RES-CLF) [46] as our certificate function. We also set $u_{\text{ref}}(x) \equiv 0$ since this naturally captures the objective of minimizing the energy spent to produce the motor torques. In order to construct RES-CLFs, we first input-output linearize the continuous dynamics of the system by defining the output functions:

$$y(q) = q_{2:5} - y_d(\theta(q)), \tag{34}$$

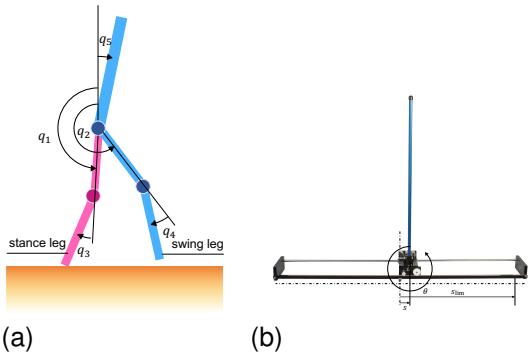

Fig. 3: (a) The configuration of the planar five-link bipedal robot RABBIT [74] (b) Cart-pole experiment setup based on Quanser Linear Servo Base Unit with Inverted Pendulum [76].

where $\theta(q)$ is a variable that defines the phase along the gait, which monotonically increases within each walking step, and $y_d(\cdot)$ is a desired gait represented by a Bezier polynomial, generated offline using the Fast Robot Optimization and Simulation Toolkit (FROST) [75]. We can then decompose the state of the system into the transverse coordinates $\xi = [y\ \dot{y}]^\top \in \mathbb{R}^8$ and the zero coordinates $\eta = [\theta(q)\ \dot{\theta}(q)] \in \mathbb{R}^2$. After applying the input-output linearization, we can represent the transverse dynamics as:

$$\dot{\xi} = f(\xi, \eta) + g(\xi, \eta)\mu, \tag{35}$$

where μ is the virtual input. By stabilizing ξ to zero, we enforce the joint trajectory to converge to the desired stable walking gait defined by $y_d(\theta(q))$.

Model uncertainty is introduced in the simulation by scaling the mass and inertia values of the robot by a factor of 2, which poses a challenge for the controller to maintain stability during walking. Note that a payload is one of the most common sources of model uncertainty for legged robots in practical applications. As illustrated in Figure 4, while the oracle CLF-QP (blue), which assumes access to the true plant dynamics, successfully completes fifteen steps, the nominal model-based CLF-QP (magenta), which is unaware of the change in mass and inertia, fails to stabilize the robot and it eventually falls down during the fourteenth step. This observation motivates the use of the GP-CLF-SOCP controller.

We collect data points represented as $\bar{x}_j = ([\xi_j, \eta_j], \mu_j)$ since we aim to learn the effect of model uncertainty in the transverse dynamics (35). The dataset is collected in an episodic learning fashion, similar to our previous work [7]. The nominal model-based CLF-QP is run in the first episode to create an initial dataset for the GP regression. Following this, the GP-CLF-SOCP is executed, and the data collected from the new trajectory is iteratively added to the dataset. For the GP-CLF-SOCP, we initially use the full dataset; however, when the execution time of the SOCP controller approaches the limit of the target sampling time, we activate the data selection algorithm. It is essential to acknowledge that high-dimensional systems are more susceptible to the out-of-distribution problem, as data is inherently more scarce. To address this challenge, we introduce perturbations to the reset map at every impact event and create variations in the control

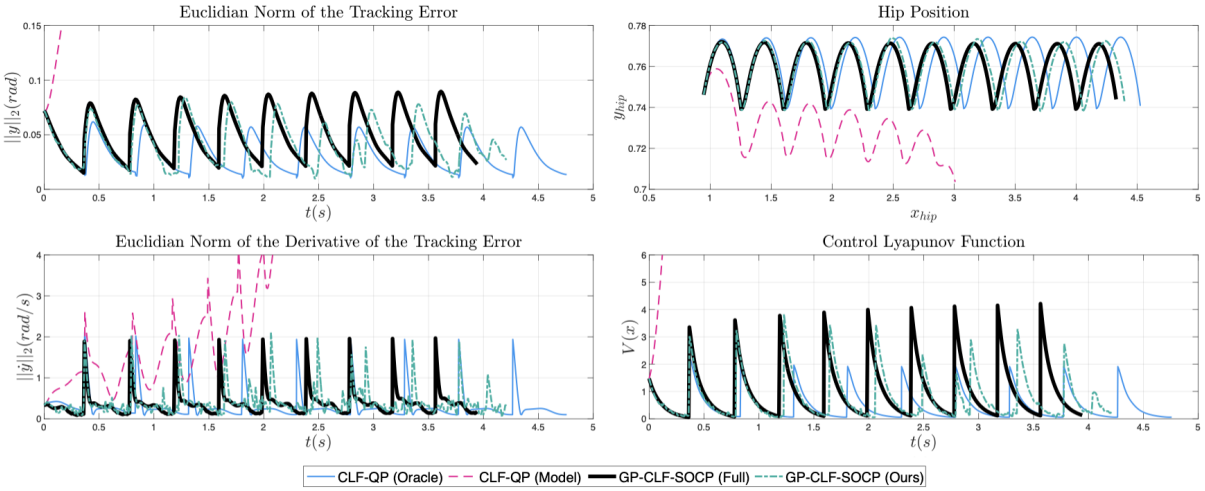

Fig. 4: Simulation results of RABBIT achieving stable walking under various controllers: the nominal model-based CLF-QP (magenta), the oracle CLF-QP (blue), GP-CLF-SOCP (Full) (black), and GP-CLF-SOCP (Ours) (green). The left column depicts histories of the Euclidean norm of the tracking error y and its time derivative $\dot{y}$ with respect to the reference gait. The right column shows the evolution of the hip's vertical position from the ground and the value of the CLF $V(x)$.

policies executed in each episode, for example, by altering the number of M, in order to enhance the dataset's coverage. As a result, we obtain a comprehensive dataset comprising 12,765 (N) data points.

When assuming the ability to deploy the GP-CLF-SOCP (Full) at a sampling rate of 40Hz (25ms), it can achieve fifteen successful steps without falling, as shown in Figure 4 (black). However, this would not be achievable in reality, as the average execution time of the controller using the full dataset is 238.3ms, which significantly exceeds the target sampling time. Instead, we employed our main data selection algorithm to choose 30 (M) data points from the full dataset. As demonstrated in Table I, this algorithm significantly reduced the execution time to an average of 22.9ms.

As shown in Figure 4, the GP-CLF-SOCP (Ours) (green) enables the robot to successfully complete fifteen steps. This is further evidenced by the CLF and tracking error plot, where the controller consistently and exponentially stabilizes the tracking error close to zero after the repeated state resets. It is worth noting that the resulting walking gait of the GP-CLF-SOCP controller differs from the oracle controller, as the SOCP controller chooses control inputs that are robust to the prediction uncertainty. Consequently, the controller behaves more conservatively; in this case, it leads to a slightly faster walking gait than that of the oracle CLF-QP controller.

C. Hardware Experiment: Cart-pole System

The importance of the method presented in this work is most notable for real hardware systems, as we can use the data collected from the real system to account for the inevitable inaccuracies that even our best possible mathematical description of its dynamics might suffer from. This is precisely what is observed in the experiment we conducted on a Quanser Linear Servo Base Unit with Inverted Pendulum [76] hardware (Figure 3 (b)). This cart-pole system consists of a linearly-actuated cart and an unactuated pendulum. The state of the system can be described as $x = [s, \dot{s}, \theta, \dot{\theta}] \in \mathbb{R}^4$, where s and $\dot{s}$ are the cart's position and velocity, and θ and $\dot{\theta}$ correspond to the pole's relative angle with respect to the upright position and its angular velocity. The control input $u \in \mathbb{R}$ is the voltage applied to the linear actuator of the cart.

TABLE I: Total execution time (data selection, GP inference, numerical optimization) of the GP-CF-SOCP controller with different datasets in the RABBIT simulation and the Cart-pole experiment. Mean and standard deviations are in milliseconds.

	GP-CF-SOCP (Ours)			**GP-CF-SOCP (Full)**		
System	mean	stdev	M	mean	stdev	N
RABBIT	22.9	4.4	30	238.3	9.4	12765
Cart-Pole	11.8	0.75	40	60.4	4.1	6957

The control objective of this experiment is to swing-up the pole to the upright position and balance it at the top, while respecting a safety constraint on the cart's position, given as $|s| \leq s_{\text{lim}} = 0.35$m. In particular, this constraint is placed to avoid the cart from colliding against the limits of the linear guide. The CBF we designed, which is then used as the certificate function, is based on the exponential CBF design methods for high relative-degree constraints [77]; in our case, the original cart position constraint has a relative degree of two. This results in a CBF expressed as

$$C(x) = -2s\dot{s} + k(s_{\text{lim}}^2 - s^2). \tag{36}$$

The zero-level set of the CBF is depicted in red in the left plot of Figure 5.

For the swing-up task, we design a reference policy u_{ref}, which is a hybrid controller that switches between an energy-based feedback controller that increases the total energy of the system until it matches the level of the potential energy of the unstable equilbrium, and a stabilizing controller to which the system switches at the vicinity of the equilibrium.

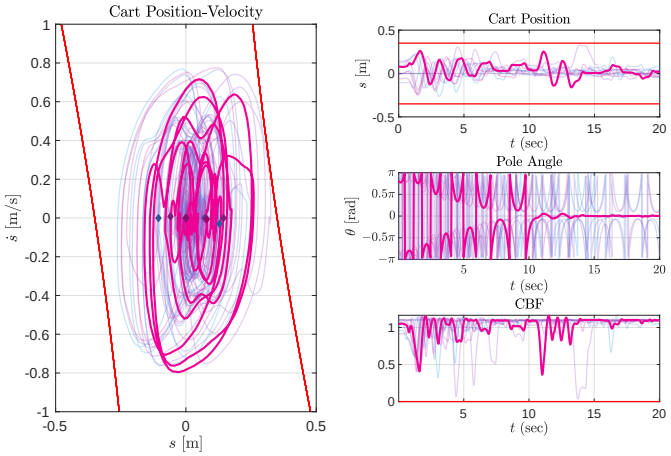

Fig. 5: 10 episodes of the cart-pole experiment under GP-CBF-SOCP (Ours). We highlight one of the ten trajectories in magenta with thick curves and the rest in thin transparent curves. On the left is the phase plot of the trajectories in the cart position and velocity space (s vs $\dot{s}$), where the region between the red curves indicates the zero-super level set of the CBF. The diamond markers indicate the initial states of the trajectories. No trajectory exits the zero-super level set of the CBF. On the right are the plots of cart position(s), pole angle (θ), and CBF value of the trajectories in time. The highlighted trajectory successfully swings up the pole while maintaining the safety constraint.

We then apply the reference policy filtered by the nominal model-based CBF-QP certifying filter. For the nominal model, we use the high-fidelity dynamics model provided by the manufacturer [76]. We apply the computed filtered control input to the actual system every 25ms, which is the target sampling time for the real-time execution of the controller. Even though the provided dynamics model tries to capture many of the complex nonlinearities present in the system, we observe that, when deployed on the real system, the nominal model-based CBF-QP still fails to satisfy the safety constraints at several trials and the CBF becomes negative.

This motivates us to employ the GP-CBF-SOCP certifying filter to achieve the swing-up task while adhering to the cart position limit, after learning the effect of model uncertainty from the data. In order to maintain feasibility of the GP-CBF-SOCP filter, the dataset must sufficiently cover the state and control input space where the system operates. We collect these data points in an episodic fashion. As more data points are aggregated, the GP inference takes longer, eventually exceeding the 25ms limit of our sampling time. Thus, we conduct the episodic procedure twice, each collecting nine trajectories, and then combine the two datasets into the full dataset. With the full dataset comprising 6957 (N) data points, we observe that the GP-CBF-SOCP controller takes too long to perform the inference, causing an average 60.4ms execution time (Table I), which does not meet the target sampling rate requirement. This effect is evident in the experiment, as the cart-pole fails to swing up properly due to the delay.

On the contrary, using our data selection algorithm with 40 (M) points selected online, the total execution time becomes much smaller, resulting in an average of 11.8ms. Over 10 experiments using our main algorithm and the GP-CBF-SOCP, we achieve 100% constraint satisfaction. These trajectories are shown in Figure 5. Although not all of these experiments result in a successful balance at the upright position within the allocated 20 seconds (achieved in 6 out of 10 experiments), the GP-CBF-SOCP successfully prioritizes safety over performance, ensuring the cart never exits the defined limits imposed by the CBF-based certifying constraint. In the video showcasing the results in video link[1], it is clear that the learned certifying constraint forces the cart to drop the pole when it approaches the position limit. Moreover, we demonstrate that even when an external user introduces disturbances by pushing the pole, the system remains safe.

VII. Conclusion

In this study, we introduce a runtime-efficient data-driven certifying filter approach applicable to real-time, complex robotic systems with uncertain dynamics that typically require large datasets for learning certified control laws. We achieved this by creating a nonparametric learning-based SOCP filter with significantly improved time complexity, transitioning from quadratic to linear with respect to dataset size, utilizing a novel online data selection algorithm. This algorithm generates a dataset most relevant to the desired certification of the system, grounded in a theoretical analysis that confirms its approximate optimality under reasonable dataset assumptions. The effectiveness of our algorithm is demonstrated in securing the safety of a real-world cart-pole swing-up task and maintaining stable locomotion for a five-link bipedal walker under significant mass uncertainty, as exhibited in the RABBIT simulation.

Our investigation into the quantification of information from individual data points for the certifying filter establishes a foundation for a more profound understanding of the relationship between data and certifying control laws. Potential future research directions include incorporating data selection objectives during the data collection process for sample-efficient learning and expanding our framework to address systems with multiple certifying constraints.

Acknowledgment

We would like to thank Andrew J. Taylor, Victor D. Dorobantu, and Alonso Marco for insightful discussions. This work is supported by DARPA Assured Autonomy Grant No. FA8750-18-C-0101, the NASA University Leadership Initiative Grant No. 80NSSC20M0163, and the National Science Foundation Grant CMMI-1944722. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any aforementioned organizations.

[1] https://youtu.be/eOxcL0pAo_o

APPENDIX

A. Proof of Lemma 1

For notational convenience, we will drop $(\cdot|\mathbb{D}_N)$. From (17), GP-CF-SOCP is feasible under $u = \alpha' \widehat{L_g C}(x)^\top$ if

$$c\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, c\alpha \widehat{L_g C}(x)^\top\right) \geq -\left(\widehat{L_f C}(x) + \gamma(C(x))\right),$$

where $c := \alpha'/\alpha > 1$. First, we compare $\sigma\left(x, c\alpha \widehat{L_g C}(x)^\top\right)$ and $\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)$ as below:

$$\begin{aligned}
&\frac{1}{c^2}\sigma^2\left(x, c\alpha \widehat{L_g C}(x)^\top\right) \\
&= \frac{1}{c^2}[1 \;\; c\alpha \widehat{L_g C}(x)]\Sigma(x) \begin{bmatrix} 1 \\ c\alpha \widehat{L_g C}(x)^\top \end{bmatrix} \\
&= [1/c \;\; \alpha \widehat{L_g C}(x)]\Sigma(x) \begin{bmatrix} 1/c \\ \alpha \widehat{L_g C}(x)^\top \end{bmatrix} \\
&= [1 \;\; \alpha \widehat{L_g C}(x)]\Sigma(x) \begin{bmatrix} 1 \\ \alpha \widehat{L_g C}(x)^\top \end{bmatrix} - \begin{bmatrix} 1 - \frac{1}{c^2} & 0 \end{bmatrix} \Sigma(x) \begin{bmatrix} 1 - \frac{1}{c^2} \\ 0 \end{bmatrix} \\
&= \sigma^2\left(x, \alpha \widehat{L_g C}(x)\right) - \left(1 - \frac{1}{c^2}\right)^2 \Sigma(x)_{[1,1]}
\end{aligned}$$

Thus,

$$\sigma^2\left(x, c\alpha \widehat{L_g C}(x)^\top\right) = c^2 \left(\sigma^2\left(x, \alpha \widehat{L_g C}(x)^\top\right) - \left(1 - \frac{1}{c^2}\right)^2 \Sigma(x)_{[1,1]} \right)$$

Using this expression, we can check that

$$\begin{aligned}
&\frac{c\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, c\alpha \widehat{L_g C}(x)^\top\right)}{c\left(\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)\right)} \\
&= \frac{\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sqrt{\sigma^2\left(x, \alpha \widehat{L_g C}(x)^\top\right) - \left(1 - \frac{1}{c^2}\right)^2 \Sigma(x)_{[1,1]}}}{\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)} \\
&> 1.
\end{aligned}$$

Finally, since $\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)$ is strictly positive from (23), by taking c satisfying

$$c \geq \frac{-\left(\widehat{L_f C}(x) + \gamma(C(x))\right)}{\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)},$$

we get

$$\begin{aligned}
&c\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, c\alpha \widehat{L_g C}(x)^\top\right) \\
&> c\left(\alpha \left\| \widehat{L_g C}(x) \right\|^2 - \beta\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right)\right) \\
&\geq -\left(\widehat{L_f C}(x) + \gamma(C(x))\right),
\end{aligned}$$

which completes the proof. □

B. Proof of Lemma 2

For notational convenience, we will drop $(\cdot|\mathbb{D}_M)$. We begin the proof by noting that

$$\begin{aligned}
&\lim_{\alpha\to\infty} \frac{1}{\alpha} \begin{bmatrix} \mathbf{k}_{*1}(x, \alpha \widehat{L_g C}(x)^\top) \\ \vdots \\ \mathbf{k}_{*M}(x, \alpha \widehat{L_g C}(x)^\top) \end{bmatrix} \\
&= \lim_{\alpha\to\infty} \frac{1}{\alpha} \begin{bmatrix} k_f(x_1, x_1) + \mathbf{k}^u_{*1}(x, \alpha \widehat{L_g C}(x)^\top) \\ \vdots \\ k_f(x_M, x_M) + \mathbf{k}^u_{*M}(x, \alpha \widehat{L_g C}(x)^\top) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{k}^u_{*1}(x, \widehat{L_g C}(x)^\top) \\ \vdots \\ \mathbf{k}^u_{*M}(x, \widehat{L_g C}(x)^\top) \end{bmatrix}. \qquad (37)
\end{aligned}$$

Thus,

$$\begin{aligned}
&\arg\min_{\mathbb{D}_M} \lim_{\alpha\to\infty} \frac{1}{\alpha}\sigma\left(x, \alpha \widehat{L_g C}(x)^\top\right) \\
&= \arg\min_{\mathbb{D}_M} \frac{1}{\alpha^2}\sigma^2\left(x, \alpha \widehat{L_g C}(x)^\top\right) \\
&= \arg\max_{\mathbb{D}_M} \lim_{\alpha\to\infty} \frac{1}{\alpha^2} \left[\mathbf{k}_{*1}(x, \alpha \widehat{L_g C}(x)^\top) \cdots \mathbf{k}_{*M}(x, \alpha \widehat{L_g C}(x)^\top)\right] \\
&\qquad (K_{\mathbb{D}_M} + \sigma_n^2 I)^{-1} \begin{bmatrix} \mathbf{k}_{*1}(x, \alpha \widehat{L_g C}(x)^\top) \\ \vdots \\ \mathbf{k}_{*M}(x, \alpha \widehat{L_g C}(x)^\top) \end{bmatrix} \text{(from (22))} \\
&= \arg\max_{\mathbb{D}_M} \left[\mathbf{k}^u_{*1}(x, \widehat{L_g C}(x)^\top) \;\; \cdots \;\; \mathbf{k}^u_{*M}(x, \widehat{L_g C}(x)^\top)\right] \\
&\qquad (K_{\mathbb{D}_M} + \sigma_n^2 I)^{-1} \begin{bmatrix} \mathbf{k}^u_{*1}(x, \widehat{L_g C}(x)^\top) \\ \vdots \\ \mathbf{k}^u_{*M}(x, \widehat{L_g C}(x)^\top) \end{bmatrix} \text{(from (37))},
\end{aligned}$$

which is precisely the objective function appearing in the Lemma. □

C. Proof of Theorem 1

For notational convenience, we use the subscript ij to indicate the (i,j)-th element of a matrix. We first present a few lemmas that will be used in the proof.

Lemma 3. Let $C = (c_{ij}) \in \mathbb{R}^{n\times n}$ be a non-negative matrix. Then, the maximal eigenvalue of C is upper bounded by its maximal row sum, that is,

$$\lambda_{\max}(C) \leq \max_i \sum_{j=1}^n c_{ij}. \qquad (38)$$

Proof. This is a corollary of Perron-Frobenius Theorem for nonnegative matrices [78, Ch.8]. □

Lemma 4 (Weyl's Inequality)**.** Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices. Then,

$$\lambda_{\min}(A + B) \geq \lambda_{\min}(A) + \lambda_{\min}(B).$$

Lemma 5. Let $S = (s_{ij}) \in \mathbb{R}^{n\times n}$ be a square matrix whose diagonal entities satisfy $s_{ii} = 1$, and whose off-diagonal entities satisfy $-1 \leq s_{ij} \leq 0$ for all $i \neq j$. Let

$\bar{S} = (\bar{s}_{ij}) \in \mathbb{R}^{n\times n}$ be a matrix whose diagonal entities are all one, and whose off-diagonal entities are $\bar{s}_{ij} = \pm s_{ij}$, where the signs can be arbitrary. Then, if S is positive definite, $\bar{S}$ is also positive definite.

Proof. This can be proved by induction. The case when $n = 1$ is trivial since there is no off-diagonal term.

Assume the lemma holds for $n = k$, that is, if S_k and $\bar{S}_k$ are constructed to satisfy the statement in the lemma, $S_k \succ 0 \Rightarrow \bar{S}_k \succ 0$ holds.

Next, consider

$$S_{k+1} = \begin{bmatrix} S_k & p_k \\ p_k^\top & 1 \end{bmatrix} \succ 0,$$

where $p_k = [s_{1(k+1)} \ \cdots \ s_{k(k+1)}]^\top$, and $-1 \le s_{i(k+1)} \le 0$ for $i = 1, \cdots, k$. Let $\bar{S}_{k+1}$ constructed according to the statement in the lemma as

$$\bar{S}_{k+1} = \begin{bmatrix} \bar{S}_k & \bar{p}_k \\ \bar{p}_k^\top & 1 \end{bmatrix}.$$

Since S_{k+1} is positive definite, by Schur complement lemma, the following holds.

$$S_k \succ 0, \ p_k^\top S_k^{-1} p_k < 1.$$

Note that $\bar{S}_k \succ 0$ holds due to the assumption of the induction. Define

$$T_k = I - S_k = \begin{bmatrix} 0 & & s_{ij} \\ & \ddots & \\ s_{ji} & & 0 \end{bmatrix}, \ \bar{T}_k = I - \bar{S}_k = \begin{bmatrix} 0 & & \bar{s}_{ij} \\ & \ddots & \\ \bar{s}_{ji} & & 0 \end{bmatrix}.$$

Then

$$\begin{aligned} \bar{p}_k^\top \bar{S}_k^{-1} \bar{p}_k =& \bar{p}_k^\top (I - \bar{T}_k)^{-1} \bar{p}_k \\ =& \sum_{t=0}^{\infty} \bar{p}_k^\top \bar{T}_k^t \, \bar{p}_k \le \sum_{t=0}^{\infty} p_k^\top T_k^t \, p_k \\ =& p_k^\top (I - T_k)^{-1} p_k = p_k^\top S_k^{-1} p_k < 1. \end{aligned}$$

Since $\bar{S}_k \succ 0$ and $\bar{p}_k^\top \bar{S}_k^{-1} \bar{p}_k < 1$ holds, by Schur complement lemma, $\bar{S}_{k+1}$ is positive definite. This shows that the lemma holds for $n = k + 1$. The lemma is proved by induction. □

Presented next is the main Proof of Theorem 1. We will drop (x, u) from $\mathbf{k}_{*i}^u$ and n_i, and $(\cdot|\mathbb{D}_M)$ for notational convenience. We want to prove

$$\begin{aligned} & [\mathbf{k}_{*1}^u \ \cdots \ \mathbf{k}_{*M}^u] \left(K_{\mathbb{D}_M} + \sigma_n^2 I\right)^{-1} \begin{bmatrix} \mathbf{k}_{*1}^u \\ \vdots \\ \mathbf{k}_{*M}^u \end{bmatrix} \\ & \ge \frac{1-\epsilon}{1+\epsilon(M-2)} \sum_{i=1}^{M} n_i^2 \end{aligned} \tag{39}$$

$$\Leftrightarrow \ [\mathbf{k}_{*1}^u \ \cdots \ \mathbf{k}_{*M}^u] \left(K_{\mathbb{D}_M} + \sigma_n^2 I\right)^{-1} \begin{bmatrix} \mathbf{k}_{*1}^u \\ \vdots \\ \mathbf{k}_{*M}^u \end{bmatrix} \tag{40}$$

$$\ge \frac{1-\epsilon}{1+\epsilon(M-2)} \times \tag{41}$$

$$[\mathbf{k}_{*1}^u \ \cdots \ \mathbf{k}_{*M}^u] \operatorname{Diag}\left(\begin{bmatrix} \frac{1}{\mathbf{k}_1} \cdots \frac{1}{\mathbf{k}_M} \end{bmatrix}\right) \begin{bmatrix} \mathbf{k}_{*1}^u \\ \vdots \\ \mathbf{k}_{*M}^u \end{bmatrix}.$$

It is sufficient to prove that

$$\left(K_{\mathbb{D}_M} + \sigma_n^2 I\right)^{-1} \succeq \frac{1-\epsilon}{1+\epsilon(M-2)} \operatorname{Diag}\left(\begin{bmatrix} \frac{1}{\mathbf{k}_1} \cdots \frac{1}{\mathbf{k}_M} \end{bmatrix}\right),$$

and this is equivalent to

$$\frac{1+\epsilon(M-2)}{1-\epsilon} \operatorname{Diag}\left([\mathbf{k}_1 \ \cdots \ \mathbf{k}_M]\right) - \left(K_{\mathbb{D}_M} + \sigma_n^2 I\right) \succeq 0. \tag{42}$$

We have

$$\begin{aligned} & \frac{1+\epsilon(M-2)}{1-\epsilon} \operatorname{Diag}\left([\mathbf{k}_1 \ \cdots \ \mathbf{k}_M]\right) - \left(K_{\mathbb{D}_M} + \sigma_n^2 I\right) \\ =& \begin{bmatrix} \frac{\epsilon(M-1)}{1-\epsilon}\mathbf{k}_1 - \sigma_n^2 & & -\mathbf{k}_{ij} \\ & \ddots & \\ -\mathbf{k}_{ji} & & \frac{\epsilon(M-1)}{1-\epsilon}\mathbf{k}_M - \sigma_n^2 \end{bmatrix} \\ =& \operatorname{Diag}\left(\sqrt{\mathbf{k}_1}, \cdots, \sqrt{\mathbf{k}_M}\right) A \operatorname{Diag}\left(\sqrt{\mathbf{k}_1} \ \cdots \ \sqrt{\mathbf{k}_M}\right), \end{aligned} \tag{43}$$

where

$$A := \begin{bmatrix} \frac{\epsilon(M-1)}{1-\epsilon} - \frac{\sigma_n^2}{\mathbf{k}_1} & & -\frac{\mathbf{k}_{ij}}{\sqrt{\mathbf{k}_i \mathbf{k}_j}} \\ & \ddots & \\ -\frac{\mathbf{k}_{ji}}{\sqrt{\mathbf{k}_j \mathbf{k}_i}} & & \frac{\epsilon(M-1)}{1-\epsilon} - \frac{\sigma_n^2}{\mathbf{k}_M} \end{bmatrix}.$$

Thus, it is sufficient to prove that A is positive semidefinite. By Lemma 4,

$$\lambda_{\min}(A) \ge \lambda_{\min}(\epsilon(M-1)\bar{S}) + \lambda_{\min}\left(A - \epsilon(M-1)\bar{S}\right),$$

where

$$\bar{S} := \begin{bmatrix} 1 & & -\frac{1}{\epsilon(M-1)} \frac{\mathbf{k}_{ij}}{\sqrt{\mathbf{k}_i \mathbf{k}_j}} \\ & \ddots & \\ -\frac{1}{\epsilon(M-1)} \frac{\mathbf{k}_{ji}}{\sqrt{\mathbf{k}_j \mathbf{k}_i}} & & 1 \end{bmatrix}.$$

Note that

$$\begin{aligned} & A - \epsilon(M-1)\bar{S} = \\ & \operatorname{Diag}\left(\frac{\epsilon^2(M-1)}{1-\epsilon} - \frac{\sigma_n^2}{\mathbf{k}_1}, \cdots, \frac{\epsilon^2(M-1)}{1-\epsilon} - \frac{\sigma_n^2}{\mathbf{k}_M}\right), \end{aligned}$$

and from (30),

$$\frac{\epsilon^2(M-1)}{1-\epsilon} - \frac{\sigma_n^2}{\mathbf{k}_i} \ge 0$$

for all $i = 1, \cdots, M$, thus, $A - \epsilon(M-1)\bar{S}$ is positive semidefinite. Therefore, it is now sufficient to prove that $\bar{S}$ is positive semidefinite.

We define

$$C := \begin{bmatrix} 0 & & \frac{1}{\epsilon(M-1)} \frac{|\mathbf{k}_{ij}|}{\sqrt{\mathbf{k}_i k_j}} \\ & \ddots & \\ \frac{1}{\epsilon(M-1)} \frac{|\mathbf{k}_{ji}|}{\sqrt{\mathbf{k}_j k_i}} & & 0 \end{bmatrix}. \quad (44)$$

By applying Lemma 3 to C which is non-negative, and by using condition (29):

$$\lambda_{\max}(C) \leq \max_i \sum_{j=1, j\neq i}^{M} \frac{1}{\epsilon(M-1)} \frac{|\mathbf{k}_{ij}|}{\sqrt{\mathbf{k}_i \mathbf{k}_j}} < \frac{1}{\epsilon(M-1)} \epsilon(M-1) = 1.$$

Thus, we have $\lambda_{\max}(C) < 1$. By applying Lemma 4, we have

$$\lambda_{\min}(I - C) \geq \lambda_{\min}(I) + \lambda_{\min}(-C) = 1 - \lambda_{\max}(C) > 0.$$

Thus, $S := I - C$ is positive definite. Note that $\bar{S}$ and S satisfy the conditions in Lemma 5 since $0 \leq \frac{1}{\epsilon(M-1)} \frac{|\mathbf{k}_{ij}|}{\sqrt{\mathbf{k}_i \mathbf{k}_j}} < \frac{1}{M-1} \leq 1$ from (29). Thus, by Lemma 5, $\bar{S}$ is positive definite. □

REFERENCES

[1] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer *et al.*, "Algorithms for verifying deep neural networks," *Foundations and Trends® in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021.

[2] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," *Advances in neural information processing systems*, vol. 32, 2019.

[3] S. Tonkens, A. Toofanian, Z. Qin, S. Gao, and S. Herbert, "Patching neural barrier functions using hamilton-jacobi reachability," *arXiv preprint arXiv:2304.09850*, 2023.

[4] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

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

[6] 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.

[7] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty," in *IEEE Conference on Decision and Control*, 2021, pp. 6762–6769.

[8] 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.

[9] F. Castañeda, J. J. Choi, W. Jung, B. Zhang, C. J. Tomlin, and K. Sreenath, "Probabilistic safe online learning with control barrier functions," *arXiv preprint arXiv:2208.10733*, 2022.

[10] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3861–3876, 2017.

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

[12] 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," in *IEEE Conference on Decision and Control*, 2021, pp. 6469–6476.

[13] V. Dhiman, M. J. Khojasteh, M. Franceschetti, and N. Atanasov, "Control barriers in bayesian learning of system dynamics," *IEEE Transactions on Automatic Control*, 2021.

[14] L. Brunke, S. Zhou, and A. P. Schoellig, "Barrier bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems," in *Learning for Dynamics and Control*, 2022, pp. 881–892.

[15] A. R. Kumar, S. Liu, J. F. Fisac, R. P. Adams, and P. J. Ramadge, "Probf: learning probabilistic safety certificates with barrier functions," in *Workshop on Safe and Robust Control of Uncertain Systems at the 35th Conference on Neural Information Processing Systems*, 2021.

[16] M. Greeff, A. W. Hall, and A. P. Schoellig, "Learning a stability filter for uncertain differentially flat systems using gaussian processes," in *IEEE Conference on Decision and Control*, 2021, pp. 789–794.

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

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

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

[20] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6814–6821.

[21] Q. Nguyen and K. Sreenath, "L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs," in *American Control Conference*, Chicago, IL, July 2015, pp. 862–867.

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

[23] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Robust adaptive control barrier functions: An adaptive and data-driven approach to safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2020.

[24] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control lyapunov functions for uncertain robotic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 6878–6884.

[25] 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.

[26] 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.

[27] K. Kazemian and S. Dean, "Random features approximation for fast data-driven control," in *NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems*, 2023.

[28] 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.

[29] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for gaussian process regression with application to safe control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[30] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[31] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[32] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies." *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.

[33] F. Pukelsheim, *Optimal design of experiments*. SIAM, 2006.

[34] H. Kim, J. Kim, Y. Jeong, S. Levine, and H. O. Song, "Emi: Exploration with mutual information," in *International Conference on Machine Learning*. PMLR, 2019.

[35] T. Alpcan and I. Shames, "An information-based learning approach to dual control," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2736–2748, 2015.

[36] E. Contal, V. Perchet, and N. Vayatis, "Gaussian process optimization with mutual information," in *International Conference on Machine Learning*. PMLR, 2014, pp. 253–261.

[37] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE conference on decision and control (CDC)*. IEEE, 2018, pp. 6059–6066.

[38] A. Lederer, A. Capone, J. Umlauft, and S. Hirche, "How training data impacts performance in learning-based control," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 905–910, 2020.

[39] A. Capone, A. Lederer, J. Umlauft, and S. Hirche, "Data selection for multi-task learning under dynamic constraints," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 959–964, 2020.

[40] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *arXiv preprint arXiv:2309.05837*, 2023.

[41] K. L. Hobbs, M. L. Mote, M. C. Abate, S. D. Coogan, and E. M. Feron, "Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems," *IEEE Control Systems Magazine*, vol. 43, no. 2, pp. 28–65, 2023.
[42] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, pp. 1–19, 2023.
[43] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human-robot interactions," in *Dynamic Systems and Control Conference*, vol. 46209. American Society of Mechanical Engineers, 2014.
[44] T. Wei and C. Liu, "Safe control algorithms using energy functions: A unified framework, benchmark, and new directions," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 238–243.
[45] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Science & Business Media, 2013, vol. 10.
[46] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.
[47] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural lyapunov-barrier functions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1724–1735.
[48] Z. Qin, D. Sun, and C. Fan, "Sablas: Learning safe control for black-box dynamical systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1928–1935, 2022.
[49] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 3699–3704.
[50] L. Lindemann, A. Robey, L. Jiang, S. Tu, and N. Matni, "Learning robust output control barrier functions from safe expert demonstrations," *arXiv preprint arXiv:2111.09971*, 2021.
[51] W. Jin, Z. Wang, Z. Yang, and S. Mou, "Neural certificates for safe control policies," *arXiv preprint arXiv:2006.08465*, 2020.
[52] T. Wei, S. Kang, W. Zhao, and C. Liu, "Persistently feasible robust safe control by safety index synthesis and convex semi-infinite programming," *IEEE Control Systems Letters*, vol. 7, pp. 1213–1218, 2023.
[53] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
[54] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE control systems letters*, vol. 3, no. 1, pp. 108–113, 2018.
[55] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2038–2044.
[56] C. Khazoom, D. Gonzalez-Diaz, Y. Ding, and S. Kim, "Humanoid self-collision avoidance using whole-body control with control barrier functions," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 558–565.
[57] T. G. Molnar and A. D. Ames, "Safety-critical control with bounded inputs via reduced order models," *arXiv preprint arXiv:2303.03247*, 2023.
[58] D. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.
[59] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *International Conference on Machine Learning*, 2017, p. 844–853.
[60] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Trans. on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2019.
[61] C. Fiedler, C. W. Scherer, and S. Trimpe, "Practical and rigorous uncertainty bounds for gaussian process regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 8, 2021, pp. 7439–7447.
[62] V. Tresp, "A bayesian committee machine," *Neural computation*, vol. 12, no. 11, pp. 2719–2741, 2000.
[63] R. Urtasun and T. Darrell, "Sparse probabilistic regression for activity-independent human pose inference," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.
[64] R. B. Gramacy and D. W. Apley, "Local gaussian process approximation for large computer experiments," *Journal of Computational and Graphical Statistics*, vol. 24, no. 2, pp. 561–578, 2015.
[65] A. Krause, "Optimizing sensing: Theory and applications," Ph.D. dissertation, Carnegie Mellon University, 2008.
[66] M. Daszykowski, B. Walczak, and D. Massart, "Representative subset selection," *Analytica chimica acta*, vol. 468, no. 1, pp. 91–103, 2002.
[67] K. Wei, R. Iyer, and J. Bilmes, "Submodularity in data subset selection and active learning," in *International conference on machine learning*. PMLR, 2015, pp. 1954–1963.
[68] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, vol. 86, no. 416, pp. 953–963, 1991.
[69] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.
[70] J. Umlauft, T. Beckers, A. Capone, A. Lederer, and S. Hirche, "Smart forgetting for safe online learning with gaussian processes," in *Learning for dynamics and control*. PMLR, 2020, pp. 160–169.
[71] K. Yu, L. Ji, and X. Zhang, "Kernel nearest-neighbor algorithm," *Neural Processing Letters*, vol. 15, no. 2, pp. 147–156, 2002.
[72] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability measures," in *International Workshop on Artificial Intelligence and Statistics*. PMLR, 2005, pp. 136–143.
[73] J. M. Phillips and S. Venkatasubramanian, "A gentle introduction to the kernel distance," *arXiv preprint arXiv:1103.1625*, 2011.
[74] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. Canudas-De-Wit, and J. Grizzle, "Rabbit: a testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.
[75] A. Hereid and A. D. Ames, "Frost*: Fast robot optimization and simulation toolkit," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 719–726.
[76] Quanser, "Linear servo base unit with inverted pendulum," Apr 2021. [Online]. Available: https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/
[77] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 322–328.
[78] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 71.

Jason J. Choi (Student Member, IEEE) received the B.S. degree in mechanical engineering from Seoul National University in 2019. He is currently pursuing a Ph.D. degree at University of California Berkeley in mechanical engineering. His research interests center on optimal control theories for nonlinear and hybrid systems, data-driven methods for safe control, and their applications to robotics and autonomous mobility.

Fernando Castañeda received a Ph.D. degree in mechanical engineering from the University of California, Berkeley in 2023. He is a Rafael del Pino Foundation Fellow (2020) and a "la Caixa" Foundation Fellow (2017). His research interests lie at the intersection of nonlinear control and data-driven methods, with a particular emphasis on ensuring the safe operation of high-dimensional systems in the real world.

Wonsuhk Jung (Student Member, IEEE) received the B.S. degree in Mechanical Engineering and Artificial Intelligence from Seoul National University in 2022. He is currently pursuing a Ph.D. degree at Georgia Institute of Technology in Robotics. His research focuses on leveraging optimal control, planning, and data-driven methodologies for the safe operation of contact-rich robotics platforms.

Bike Zhang (Student Member, IEEE) received the B.Eng. degree in electrical engineering and automation from Huazhong University of Science and Technology in 2017. He is currently working toward the Ph.D. degree in mechanical engineering at University of California Berkeley. His current research interests include predictive control and reinforcement learning with application to legged robotics.

Claire J. Tomlin (Fellow, IEEE) is the James and Katherine Lau Professor of Engineering and professor and chair of the Department of Electrical Engineering and Computer Sciences (EECS) at UC Berkeley. She was an assistant, associate, and full professor in aeronautics and astronautics at Stanford University from 1998 to 2007, and in 2005, she joined UC Berkeley. She works in the area of control theory and hybrid systems, with applications to air traffic management, UAV systems, energy, robotics, and systems biology. She is a MacArthur Foundation Fellow (2006), an IEEE Fellow (2010), and in 2017, she was awarded the IEEE Transportation Technologies Award. In 2019, Claire was elected to the National Academy of Engineering and the American Academy of Arts and Sciences.

Koushil Sreenath (Member, IEEE) is an associate professor of mechanical engineering, at UC Berkeley. He received a Ph.D. degree in electrical engineering and computer science and a M.S. degree in applied mathematics from the University of Michigan at Ann Arbor, MI, in 2011. He was a postdoctoral scholar at the GRASP Lab at University of Pennsylvania from 2011 to 2013 and an assistant professor at Carnegie Mellon University from 2013 to 2017. His research interest lies at the intersection of highly dynamic robotics and applied nonlinear control. He received the NSF CAREER, Hellman Fellow, Best Paper Award at the Robotics: Science and Systems (RSS), and the Google Faculty Research Award in Robotics.