

Control Barriers in Bayesian Learning of System Dynamics

Vikas Dhiman*, Mohammad Javad Khojasteh*, *Member, IEEE*, Massimo Franceschetti, *Fellow, IEEE*,
and Nikolay Atanasov, *Member, IEEE*

Abstract—This paper focuses on learning a model of system dynamics online while satisfying safety constraints. Our objective is to avoid offline system identification or hand-specified models and allow a system to safely and autonomously estimate and adapt its own model during operation. Given streaming observations of the system state, we use Bayesian learning to obtain a distribution over the system dynamics. Specifically, we propose a new matrix variate Gaussian process (MVGP) regression approach with an efficient covariance factorization to learn the drift and input gain terms of a nonlinear control-affine system. The MVGP distribution is then used to optimize the system behavior and ensure safety with high probability, by specifying control Lyapunov function (CLF) and control barrier function (CBF) chance constraints. We show that a safe control policy can be synthesized for systems with arbitrary relative degree and probabilistic CLF-CBF constraints by solving a second order cone program (SOCP). Finally, we extend our design to a self-triggering formulation, adaptively determining the time at which a new control input needs to be applied in order to guarantee safety.

Index Terms—Gaussian Process, learning for dynamics and control, high relative-degree system safety, control barrier function, self-triggered safe control

SUPPLEMENTARY MATERIAL

Software and videos supplementing this paper are available at: https://vikasdhiman.info/Bayesian_CBF

I. INTRODUCTION

Unmanned vehicles and other cyber-physical systems [2], [3] promise to transform many aspects of our lives, including transportation, agriculture, mining, and construction. Successful use of autonomous systems in these areas critically depends on safe adaptation in changing operational conditions. Existing systems, however, rely on brittle hand-designed dynamics models and safety rules that often fail to account for both the complexity and uncertainty of real-world operation. Recent work [4]–[19] has demonstrated that learning-based system identification and control techniques may be successful at complex tasks and control objectives. However, two critical considerations for applying these techniques onboard autonomous systems remain less explored: *learning* online,

relying on streaming data, and guaranteeing *safe* operation, despite the estimation errors inherent to learning algorithms. For example, consider steering a Ackermann-drive vehicle, whose dynamics are not perfectly known, safely to a goal location. Not only does the Ackermann vehicle need to *learn* a distribution over its dynamics from state and control observations but also account for the variance of the dynamics to *guarantee safety* while executing its actions.

Several approaches have been proposed in the literature to guarantee safety of dynamical systems. Motivated by the utility of Lyapunov functions for certifying stability properties, [20]–[29] proposed *control barrier functions* (CBFs) as a tool to enforce safety properties in dynamical systems. A CBF certifies whether a control policy achieves forward invariance of a *safe set* \mathcal{C} by evaluating if the system trajectory remains away from the boundary of \mathcal{C} . A lot of the literature on CBFs considers systems with known dynamics, low relative degree, no disturbances, and time-triggered control, in which inputs are recalculated at a fixed and sufficiently small period. Time-triggered control is limiting because low frequency may lead to safety violations in-between sampling times, while high frequency leads to inefficient use of computational resources and actuators. Yang et al. [30] extend the CBF framework, for *known dynamics*, to a event-triggered setup [31]–[34] in which the longest time until a control input needs to be recomputed to guarantee safety is provided. CBF techniques can handle nonlinear control-affine systems but many existing results apply only to relative-degree-one systems, in which the first time derivative of the CBF depends on the control input. This requirement is violated by many underactuated robot systems and motivates extensions to relative-degree-two systems, such as bipedal and car-like robots. The works [35]–[38] generalized these ideas, in the case of *known dynamics*, by designing exponential control barrier function (ECBF) and high order control barrier function (HOCBF), that can handle control-affine systems with any relative degree.

Our work proposes a Bayesian learning approach for estimating the posterior distribution of the control-affine system dynamics from online data. We generalize the CBF control synthesis techniques to handle probabilistic safety constraints induced by the dynamics distribution. Our work makes the following **contributions**. First, we develop a Matrix Variate Gaussian Process (MVGP) regression to estimate the unknown system dynamics and formulate probabilistic safety and stability constraints for systems with arbitrary relative degree. Second, we show that a control policy satisfying the proposed probabilistic constraints can be obtained as the solution of a deterministic second order cone program (SOCP). The SOCP formulation depends on the mean and variance of the probabilistic safety and stability constraints, which can be obtained efficiently. Third, we extend our results to

* The first two authors contributed equally.

The material in this paper was presented in part at the 2020 Learning for Dynamics and Control Conference (L4DC) [1]. We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181 and NSF awards CNS-1446891, ECCS-1917177, and IIS-2007141.

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

M. J. Khojasteh is with Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139. (e-mail: mkhojast@mit.edu)

M. Franceschetti, and N. Atanasov are with Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093. (e-mails: {mfranceschetti, natanasov}@ucsd.edu)

a self-triggered safe control setting, adaptively determining the duration of each control input before re-computation is necessary to guarantee safety with high probability. Finally, we derive closed-form expressions for the mean and variance of the probabilistic safety and stability constraints, up to relative-degree two. This work extends our conference paper [1] by introducing a SOCP control-synthesis formulation instead of the original, possibly non-convex, quadratic program (QP) problem. This paper also extends the self-triggered control design to systems with relative degree above one, contains a complete technical treatment of the result—including the proofs of that were omitted in the conference version, and presents new evaluation results.

II. RELATED WORK

Providing safety guarantees for learning-based control techniques has received significant attention recently [39]–[46]. In particular, optimization-based control synthesis with CLF and CBF constraints has been considered for systems subject to additive stochastic disturbances [47]–[49]. Aloysius Pereira et al. [49] combine CBF constraints and forward-backward stochastic differential equations, to find a deep parameterized safe optimal control policy in the presence of additive Brownian motion in the system dynamics. Yaghoubi et al. [50] provide a stochastic CBF framework for composite systems where a component of the system evolve according to a known deterministic dynamics and the other one follows a stochastic dynamics based on the Weiner process. CBF conditions for systems with uncertain dynamics have been proposed in [51]–[58]. Fan et al. [51] study time-triggered CBF-based control synthesis for control-affine systems with relative degree one, where the input gain is known and invertible but the drift term is learned via Bayesian techniques. The authors compare the performance of Gaussian process (GP) regression [59], dropout neural networks [60], and ALPaCA [61] in constructing adaptive CLF and CBF conditions using bounds on the error with respect to a system reference model. The works in [52], [53], [55], [56] study time-triggered CBF-based control of relative-degree-one systems with additive uncertainties in the drift part of the dynamics. Wang et al. [52] use GP regression to approximate the unknown part of the 3D nonlinear dynamics of a quadrotor robot. Cheng et al. [53] propose a two-layers control policy design that integrates CBF-based control with model-free reinforcement learning (RL). Safety is ensured by bounding the worst-case deviation of the dynamics estimate from the mean, using high-confidence polytopic uncertainty bounds. Marvi and Kiumarsi [54] also consider safe model-free reinforcement learning but, instead of a two-layer policy design, the cost-to-go function is augmented with a CBF term. The works [55], [56] use adaptive CBFs to deal with parameter uncertainty. Salehi et al. [57] use Extreme Learning Machines to approximate the dynamics of a closed-loop nonlinear system (with a drift term only) subject to a barrier certificate constraint during the learning process itself.

Our work builds upon the current literature by developing a matrix variate GP regression with efficient covariance factorization to learn both the *drift term* and the *input gain* terms of a nonlinear control-affine system. The posterior distribution is used to ensure safety for systems with *arbitrary relative*

degree. Compared to previous works, our safety constraints are less conservative (probabilistic instead of worst-case) and lead to a novel SOCP formulation. We also present results for a *self-triggering design* with unknown system dynamics.

Research directions left open for future investigation include the extension of the results to a frequentist setting [62]. Following [63], where the unknown but deterministic dynamics are assumed to belong to the Reproducing Kernel Hilbert Space (RKHS), our CBF-based self-triggered controller may be redesigned in a frequentist framework. In fact, a time-triggered setup that utilizes independent GP regression for each coordinate has been recently developed in [64], where a systematic approach is utilized to compute CBFs for the learned model.

III. PROBLEM STATEMENT

Consider a control-affine nonlinear system:

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

where $\mathbf{x}(t) \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathcal{U} \subset \mathbb{R}^m$ are the system state and control input, respectively, at time t . Assume that the *drift term* $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and the *input gain* $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz and the admissible control set \mathcal{U} is convex. We study the problem of enforcing stability and safety properties for (1) when f and g are unknown and need to be estimated online, using observations of \mathbf{x} , \mathbf{u} , $\dot{\mathbf{x}}$.

A. Notation

We use bold lower-case letters for vectors (\mathbf{x}), bold capital letters for matrices (\mathbf{X}), and caligraphic capital letters for sets (\mathcal{X}). The boundary of a set \mathcal{X} is denoted by $\partial\mathcal{X}$. Let $\text{vec}(\mathbf{X}) \in \mathbb{R}^{nm}$ be the vectorization of $\mathbf{X} \in \mathbb{R}^{n \times m}$, obtained by stacking the columns of \mathbf{X} . The Kronecker product is denoted by \otimes . The Hessian and Jacobian of functions $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, respectively, are defined as:

$$\mathcal{H}_{\mathbf{x},\mathbf{y}}h(\mathbf{x},\mathbf{y}) \triangleq \begin{bmatrix} \frac{\partial^2 h}{\partial x_1 \partial y_1} & \cdots & \frac{\partial^2 h}{\partial x_1 \partial y_m} \\ \vdots & & \vdots \\ \frac{\partial^2 h}{\partial x_n \partial y_1} & \cdots & \frac{\partial^2 h}{\partial x_n \partial y_m} \end{bmatrix} \quad \mathcal{J}_{\mathbf{x}}f(\mathbf{x}) \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

The Lie derivative of $V : \mathbb{R}^n \rightarrow \mathbb{R}$ along $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is denoted by $\mathcal{L}_f V : \mathbb{R}^n \rightarrow \mathbb{R}$. The space of r times continuously differentiable functions $h : \mathcal{X} \rightarrow \mathbb{R}$ is denoted by $\mathcal{C}^r(\mathcal{X}, \mathbb{R})$.

B. Stability and Safety with Known Dynamics

We first review key results [25] on control Lyapunov functions for enforcing stability and control barrier functions for enforcing safety of control-affine systems with *known dynamics*. System stability may be asserted as follows.

Definition 1. A function $V \in \mathcal{C}^1(\mathcal{X}, \mathbb{R})$ is a *control Lyapunov function* (CLF) for the system in (1) if it is positive definite, $V(\mathbf{x}) > 0$, $\forall \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{0}\}$, $V(\mathbf{0}) = 0$, and satisfies:

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

where $\text{CLC}(\mathbf{x}, \mathbf{u}) \triangleq \mathcal{L}_f V(\mathbf{x}) + \mathcal{L}_g V(\mathbf{x})\mathbf{u} + \gamma(V(\mathbf{x}))$ is a *control Lyapunov condition* (CLC) defined for some class K function γ .

Proposition 1 (Sufficient Condition for Stability [25]). *If there exists a CLF $V(\mathbf{x})$ for system (1), then any Lipschitz continuous control policy $\pi(\mathbf{x}) \in \{\mathbf{u} \in \mathcal{U} \mid \text{CLC}(\mathbf{x}, \mathbf{u}) \leq 0\}$ asymptotically stabilizes the system.*

Let $\mathcal{C} \triangleq \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$ be a *safe set* of system states, defined implicitly by a function $h \in \mathbb{C}^1(\mathcal{X}, \mathbb{R})$. System (1) is *safe* with respect to \mathcal{C} if \mathcal{C} is *forward invariant*, i.e., for any $\mathbf{x}(0) \in \mathcal{C}$, $\mathbf{x}(t)$ remains in \mathcal{C} for all $t \geq 0$. System safety may be asserted as follows.

Definition 2. A function $h \in \mathbb{C}^1(\mathcal{X}, \mathbb{R})$ is a *control barrier function* (CBF) for the system in (1) if

$$\sup_{\mathbf{u} \in \mathcal{U}} \text{CBC}(\mathbf{x}, \mathbf{u}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \quad (3)$$

where $\text{CBC}(\mathbf{x}, \mathbf{u}) \triangleq \mathcal{L}_f h(\mathbf{x}) + \mathcal{L}_g h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x}))$ is a *control barrier condition* (CBC) defined for some extended class K_∞ function α .

Proposition 2 (Sufficient Condition for Safety [25]). *Consider a set \mathcal{C} defined implicitly by $h \in \mathbb{C}^1(\mathcal{X}, \mathbb{R})$. If h is a CBF and $\nabla h(\mathbf{x}) \neq 0$ for all \mathbf{x} when $h(\mathbf{x}) = 0$, then any Lipschitz continuous control policy $\pi(\mathbf{x}) \in \{\mathbf{u} \in \mathcal{U} \mid \text{CBC}(\mathbf{x}, \mathbf{u}) \geq 0\}$ renders the system in (1) safe with respect to \mathcal{C} .*

Prop. 1 and Prop. 2 provide sufficient conditions for a control policy $\pi(\mathbf{x})$ applied to the system in (1) to guarantee stability and safety. Note that the conditions are defined by affine constraints in \mathbf{u} . This allows the formulation of control synthesis as a quadratic program (QP) in which stability and safety properties are captured by the linear CLC and CBC constraints, respectively:

$$\begin{aligned} \pi(\mathbf{x}) \in \arg \min_{\mathbf{u} \in \mathcal{U}, \delta \in \mathbb{R}} \quad & \|\mathbf{R}(\mathbf{x})\mathbf{u}\|^2 + \lambda \delta^2 \\ \text{s.t.} \quad & \text{CLC}(\mathbf{x}, \mathbf{u}) \leq \delta, \text{CBC}(\mathbf{x}, \mathbf{u}) \geq 0, \end{aligned} \quad (4)$$

where $\mathbf{R}(\mathbf{x}) \in \mathbb{R}^{m \times m}$ is a matrix penalizing control effort and δ is a slack variable that ensures feasibility of the QP by giving preference to safety over stability, controlled by the scaling factor $\lambda > 0$. If a stabilizing control policy $\hat{\pi}(\mathbf{x})$ is already available, it may be modified minimally online to ensure safety:

$$\begin{aligned} \pi(\mathbf{x}) \in \arg \min_{\mathbf{u} \in \mathcal{U}} \quad & \|\mathbf{R}(\mathbf{x})(\mathbf{u} - \hat{\pi}(\mathbf{x}))\|^2 \\ \text{s.t.} \quad & \text{CBC}(\mathbf{x}, \mathbf{u}) \geq 0. \end{aligned} \quad (5)$$

In practice, the QPs above cannot be solved infinitely fast. Optimization is typically performed at triggering times t_k, t_{k+1}, \dots , providing control input $\mathbf{u}_k \triangleq \pi(\mathbf{x}_k)$ when the system state is $\mathbf{x}_k \triangleq \mathbf{x}(t_k)$. Ames et al. [21, Thm. 3] show that if f , g , and $\alpha \circ h$ are locally Lipschitz, then $\pi(\mathbf{x})$ and $\text{CBC}(\mathbf{x}, \pi(\mathbf{x}))$ are locally Lipschitz. Thus, for sufficiently small inter-triggering times $\tau_k \triangleq t_{k+1} - t_k$, solving (5) at $\{t_k\}_{k \in \mathbb{N}}$ ensures safety during the inter-triggering intervals $[t_k, t_{k+1})$ as well.

C. Stability and Safety with Unknown Dynamics

This work considers stability and safety for the control-affine nonlinear system in (1) when the system dynamics

$F(\mathbf{x}) \in \mathbb{R}^{n \times (1+m)}$ are *unknown*. We place a prior distribution over $\text{vec}(F(\mathbf{x}))$ using a GP [59] with mean function $\text{vec}(\mathbf{M}_0(\mathbf{x}))$ and covariance function $\mathbf{K}_0(\mathbf{x}, \mathbf{x}')$.

Our objective is to compute the posterior distribution of $\text{vec}(F(\mathbf{x}))$ using observations of the system states and controls over time and ensure stability and safety using the estimated dynamics model despite possible estimation errors.

Problem 1. Given a prior distribution on the unknown system dynamics, $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\mathbf{M}_0(\mathbf{x})), \mathbf{K}_0(\mathbf{x}, \mathbf{x}'))$, and a training set, $\mathbf{X}_{1:k} \triangleq [\mathbf{x}(t_1), \dots, \mathbf{x}(t_k)]$, $\mathbf{U}_{1:k} \triangleq [\mathbf{u}(t_1), \dots, \mathbf{u}(t_k)]$, $\dot{\mathbf{X}}_{1:k} = [\dot{\mathbf{x}}(t_1), \dots, \dot{\mathbf{x}}(t_k)]^1$, compute the posterior distribution $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{K}_k(\mathbf{x}, \mathbf{x}'))$ of $\text{vec}(F(\mathbf{x}))$ conditioned on $(\mathbf{X}_{1:k}, \mathbf{U}_{1:k}, \dot{\mathbf{X}}_{1:k})$.

Problem 2. Given a safe set $\mathcal{C} \triangleq \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$, initial state $\mathbf{x}_k \triangleq \mathbf{x}(t_k) \in \mathcal{C}$, and the distribution $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{K}_k(\mathbf{x}, \mathbf{x}'))$ of $\text{vec}(F(\mathbf{x}))$ at time t_k , choose a control input \mathbf{u}_k and triggering period τ_k such that for $\mathbf{u}(t) \triangleq \mathbf{u}_k$:

$$\mathbb{P}(\text{CBC}(\mathbf{x}(t), \mathbf{u}_k) \geq 0) \geq p_k \quad \text{for all } t \in [t_k, t_k + \tau_k), \quad (6)$$

where $\mathbf{x}(t)$ follows the dynamics in (1), and $p_k \in (0, 1)$ is a user-specified risk tolerance.

IV. MATRIX VARIATE GAUSSIAN PROCESS REGRESSION OF SYSTEM DYNAMICS

This section presents our novel Matrix Variate Gaussian Process (MVGP) solution to Problem 1. We aim to estimate the unknown drift f and input gain g of system (1) using a state-control dataset $(\mathbf{X}_{1:k}, \mathbf{U}_{1:k}, \dot{\mathbf{X}}_{1:k})$. To simplify the derivation, we assume that $\mathbf{X}_{1:k}$ and $\mathbf{U}_{1:k}$ are observed without noise, but each measurement $\dot{\mathbf{x}}(t_k)$ in $\dot{\mathbf{X}}_{1:k}$ is corrupted by zero-mean Gaussian noise $\mathcal{N}(\mathbf{0}, \mathbf{S})$ that is independent across time steps. Treating $f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$ as a single vector-valued function with input $[\mathbf{x}^\top, \mathbf{u}^\top]^\top$ may seem natural from a function approximation perspective, but this approach ignores the control-affine structure. Encoding this structure in the learning process not only increases the learning efficiency, but also ensures that the control Lyapunov and control barrier conditions remain linear in \mathbf{u} . Hence, we focus on learning the matrix-valued function $F(\mathbf{x}) = [f(\mathbf{x}) \ g(\mathbf{x})]$ defined in (1).

The simplest approach is to use $n(1+m)$ decoupled GPs for each element of $F(\mathbf{x})$. This approach ignores the dependencies among the components of $f(\mathbf{x})$ and $g(\mathbf{x})$. Furthermore, since the outputs of $f(\mathbf{x})$ and $g(\mathbf{x})$ are observed together via $\dot{\mathbf{X}}_{1:k}$, training data dimensions are still mutually correlated and cannot be treated as decoupled GPs denying the efficiency advantage. At the other extreme, treating $\text{vec}(F(\mathbf{x}))$ as a single vector-valued function and using a single GP distribution will enable high estimation accuracy but specifying an effective matrix-valued kernel function $\mathbf{K}_0(\mathbf{x}, \mathbf{x}')$ and optimizing its hyperparameters is challenging. A promising approach is offered by the Coregionalization GP (CoGP) model [65], where the kernel function is decomposed as $\mathbf{K}_0(\mathbf{x}, \mathbf{x}') \triangleq \Sigma \kappa_0(\mathbf{x}, \mathbf{x}')$ into a scalar kernel $\kappa_0(\mathbf{x}, \mathbf{x}')$ and a covariance matrix parameter $\Sigma \in \mathbb{R}^{n(1+m) \times (1+m)n}$. Estimating Σ may

¹If not available, the derivatives may be approximated via finite differences, e.g., $\dot{\mathbf{X}}_{1:k} \triangleq \left[\frac{\mathbf{x}(t_2) - \mathbf{x}(t_1)}{t_2 - t_1}, \dots, \frac{\mathbf{x}(t_{k+1}) - \mathbf{x}(t_k)}{t_{k+1} - t_k} \right]$, provided that the inter-triggering times $\{\tau_k = t_k - t_{k-1}\}_k$ are sufficiently small.

still require a lot of training data. Moreover, the matrix-times-scalar-kernel structure is not preserved in the posterior of the Coregionalization model, preventing its effective use for incremental learning when new data is received over time.

We propose an alternative factorization of $\mathbf{K}_0(\mathbf{x}, \mathbf{x}')$ inspired by the Matrix Variate Gaussian distribution [66], [67]. The strengths of our factorization are that it models the correlation among the elements of $F(\mathbf{x})$, preserves its structure in the posterior GP distribution, and has similar training and testing complexity as the decoupled GP approach.

Definition 3. The Matrix Variate Gaussian (MVG) distribution is a three-parameter distribution $\mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$ describing a random matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ with probability density function:

$$p(\mathbf{X}; \mathbf{M}, \mathbf{A}, \mathbf{B}) \triangleq \frac{\exp\left(-\frac{1}{2} \text{tr}[\mathbf{B}^{-1}(\mathbf{X} - \mathbf{M})^\top \mathbf{A}^{-1}(\mathbf{X} - \mathbf{M})]\right)}{(2\pi)^{nm/2} \det(\mathbf{A})^{m/2} \det(\mathbf{B})^{n/2}}$$

where $\mathbf{M} \in \mathbb{R}^{n \times m}$ is the mean. Up to a scaling factor, each column of \mathbf{X} has the same covariance $\mathbf{A} \in \mathbb{R}^{n \times n}$, and each row has the same covariance $\mathbf{B} \in \mathbb{R}^{m \times m}$.

Additional properties of the MVG distribution are provided in Appendix A. Note that if $\mathbf{X} \sim \mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$, then $\text{vec}(\mathbf{X}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{B} \otimes \mathbf{A})$. Based on this observation, we propose the following decomposition of the GP prior:

$$\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\mathbf{M}_0(\mathbf{x})), \mathbf{B}_0(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A}), \quad (7)$$

where $\mathbf{B}_0(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{(m+1) \times (m+1)}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$. We call such a process $F(\mathbf{x})$ with kernel structure defined in (7) a Matrix Variate Gaussian Process (MVGP).

We assume that the measurement noise covariance matrix satisfies $\mathbf{S} = \sigma^2 \mathbf{A}$ for some parameter $\sigma > 0$. This assumption on the relationship between \mathbf{S} and the dynamics row covariance \mathbf{A} is necessary to preserve the Kronecker product structure of the covariance in the GP posterior.

To simplify notation, let $\mathbf{B}_{1:k}^{1:k} \in \mathbb{R}^{k(m+1) \times k(m+1)}$ be a matrix with $k \times k$ block elements $[\mathbf{B}_{1:k}^{1:k}]_{i,j} \triangleq \mathbf{B}_0(\mathbf{x}_i, \mathbf{x}_j)$ and define $\mathcal{M}_{1:k} \triangleq [\mathbf{M}_0(\mathbf{x}_1) \ \cdots \ \mathbf{M}_0(\mathbf{x}_k)] \in \mathbb{R}^{n \times k(m+1)}$, $\mathbf{B}_{1:k}(\mathbf{x}) \triangleq [\mathbf{B}_0(\mathbf{x}, \mathbf{x}_1), \dots, \mathbf{B}_0(\mathbf{x}, \mathbf{x}_k)] \in \mathbb{R}^{(m+1) \times k(m+1)}$ and $\underline{\mathbf{u}}_{1:k} \triangleq \text{diag}(\underline{\mathbf{u}}_1, \dots, \underline{\mathbf{u}}_k) \in \mathbb{R}^{k(m+1) \times k}$. Consider an arbitrary test point \mathbf{x}_* . The train and test data are jointly Gaussian:

$$\begin{bmatrix} \text{vec}(\dot{\mathbf{X}}_{1:k}) \\ \text{vec}(F(\mathbf{x}_*)) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \text{vec}(\mathcal{M}_{1:k} \underline{\mathbf{u}}_{1:k}) \\ \text{vec}(\mathbf{M}_0(\mathbf{x}_*)) \end{bmatrix}, \begin{bmatrix} \underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}^{1:k} \underline{\mathbf{u}}_{1:k} + \sigma^2 \mathbf{I}_k & \underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}^\top(\mathbf{x}_*) \\ \mathbf{B}_{1:k}(\mathbf{x}_*) \underline{\mathbf{u}}_{1:k} & \mathbf{B}_0(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \otimes \mathbf{A} \right), \quad (8)$$

where

$$\begin{aligned} \underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}^{1:k} \underline{\mathbf{u}}_{1:k} &= \begin{bmatrix} \underline{\mathbf{u}}_1^\top \mathbf{B}_0(\mathbf{x}_1, \mathbf{x}_1) \underline{\mathbf{u}}_1 & \cdots & \underline{\mathbf{u}}_1^\top \mathbf{B}_0(\mathbf{x}_1, \mathbf{x}_k) \underline{\mathbf{u}}_k \\ \vdots & \ddots & \vdots \\ \underline{\mathbf{u}}_k^\top \mathbf{B}_0(\mathbf{x}_k, \mathbf{x}_1) \underline{\mathbf{u}}_1 & \cdots & \underline{\mathbf{u}}_k^\top \mathbf{B}_0(\mathbf{x}_k, \mathbf{x}_k) \underline{\mathbf{u}}_k \end{bmatrix} \\ \mathbf{B}_{1:k}(\mathbf{x}_*) \underline{\mathbf{u}}_{1:k} &= [\mathbf{B}_0(\mathbf{x}_1, \mathbf{x}_*) \underline{\mathbf{u}}_1 \ \cdots \ \mathbf{B}_0(\mathbf{x}_k, \mathbf{x}_*) \underline{\mathbf{u}}_k]. \end{aligned} \quad (9)$$

Applying a Schur complement to (8), we find the distribution of $\text{vec}(F(\mathbf{x}_*))$ conditioned on the training data $(\mathbf{X}_{1:k}, \mathbf{U}_{1:k}, \dot{\mathbf{X}}_{1:k})$. The posterior mean and covariance functions are provided in the next proposition.

Proposition 3. The posterior distribution of $\text{vec}(F(\mathbf{x}))$ conditioned on the training data $(\mathbf{X}_{1:k}, \mathbf{U}_{1:k}, \dot{\mathbf{X}}_{1:k})$ is a Gaussian process $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{B}_k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A})$ with parameters:

$$\begin{aligned} \mathbf{M}_k(\mathbf{x}) &\triangleq \mathbf{M}_0(\mathbf{x}) + \left(\dot{\mathbf{X}}_{1:k} - \mathcal{M}_{1:k} \underline{\mathbf{u}}_{1:k} \right) \left(\underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}(\mathbf{x}) \right)^\dagger \\ \mathbf{B}_k(\mathbf{x}, \mathbf{x}') &\triangleq \mathbf{B}_0(\mathbf{x}, \mathbf{x}') - \mathbf{B}_{1:k}(\mathbf{x}) \underline{\mathbf{u}}_{1:k} \left(\underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}(\mathbf{x}') \right)^\dagger \\ \left(\underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}(\mathbf{x}) \right)^\dagger &\triangleq \left(\underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}^{1:k} \underline{\mathbf{u}}_{1:k} + \sigma^2 \mathbf{I}_k \right)^{-1} \underline{\mathbf{u}}_{1:k}^\top \mathbf{B}_{1:k}^\top(\mathbf{x}). \end{aligned} \quad (10)$$

Proof. See Appendix B. \square

Prop. 3 shows that, for a given control input $\underline{\mathbf{u}}$, the MVGP posterior of $F(\mathbf{x})\underline{\mathbf{u}}$ is (applying Lemma 8):

$$F(\mathbf{x})\underline{\mathbf{u}} \sim \mathcal{GP}(\mathbf{M}_k(\mathbf{x})\underline{\mathbf{u}}, \underline{\mathbf{u}}^\top \mathbf{B}_k(\mathbf{x}, \mathbf{x}')\underline{\mathbf{u}} \otimes \mathbf{A}). \quad (11)$$

A key property of the MVGP model in Prop. 3 is that the posterior preserves the kernel decomposition. In the special case of, $\mathbf{B}_0(\mathbf{x}, \mathbf{x}') \triangleq \mathbf{B}_{\kappa_0}(\mathbf{x}, \mathbf{x}')$, in addition to the kernel parameters, our model requires learning of $\mathbf{B} \in \mathbb{R}^{(1+m) \times (1+m)}$ and, if the measurement noise statistics are unknown, $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\sigma \in \mathbb{R}$. Thus, the MVGP model has fewer parameters than a CoGP model, which requires $(1+m)^2 n^2$ parameters for the full covariance decomposition $\mathbf{K}_0(\mathbf{x}, \mathbf{x}') \triangleq \Sigma_{\kappa_0}(\mathbf{x}, \mathbf{x}')$. For a data set with k examples, the training computational complexity of our MVGP approach is $O(k^3)$, while the same for the CoGP model is $O(k^3 n^3)$. This is because our model inverts only a $(k \times k)$ matrix while the CoGP model inverts a $(kn \times kn)$ matrix (see Appendix C). Fig. 1 shows a comparison of the covariances obtained from an MVGP model (with $\mathbf{B}_0(\mathbf{x}, \mathbf{x}') = \mathbf{B}_{\kappa_0}(\mathbf{x}, \mathbf{x}')$) and from a CoGP model. The shapes of the MVGP covariances for different columns of $F(\mathbf{x})$ are the same by construction, while those of CoGP model may be arbitrary. Thus, the CoGP model is more expressive than our MVGP model. However, MVGP model offers considerable computational advantages especially for high-dimensional systems without significant loss in accuracy. The accuracy and computational complexity of the MVGP and CoGP models are compared in Sec. VIII-A.

V. SELF-TRIGGERED CONTROL WITH PROBABILISTIC SAFETY CONSTRAINTS

The MVGP model developed in Sec. IV addresses Problem 1 and provides a probabilistic estimate of the unknown system dynamics $F(\mathbf{x})\underline{\mathbf{u}}$ in (11). In this section, we consider Problem 2. We extend the optimization-based control synthesis approach in (4) to handle probabilistic stability and safety constraints induced by the posterior distribution of $F(\mathbf{x})\underline{\mathbf{u}}$. We focus on handling probabilistic safety constraints of the form $\mathbb{P}(\text{CBC}(\mathbf{x}, \underline{\mathbf{u}}) \geq 0) \geq p$. Since the control Lyapunov and control barrier conditions have the same form, involving the Lie derivative of a known function $V(\mathbf{x})$ or $h(\mathbf{x})$ with respect to the system dynamics, our approach can be used for stability constraints as well, as demonstrated in Sec. VIII-B. Furthermore, we develop self-triggering conditions to adaptively decide the times t_k, t_{k+1}, \dots , at which the system inputs $\underline{\mathbf{u}}$ should be recomputed in order to guarantee safety with high probability along the continuous-time system trajectory, instead of instantaneously in time.

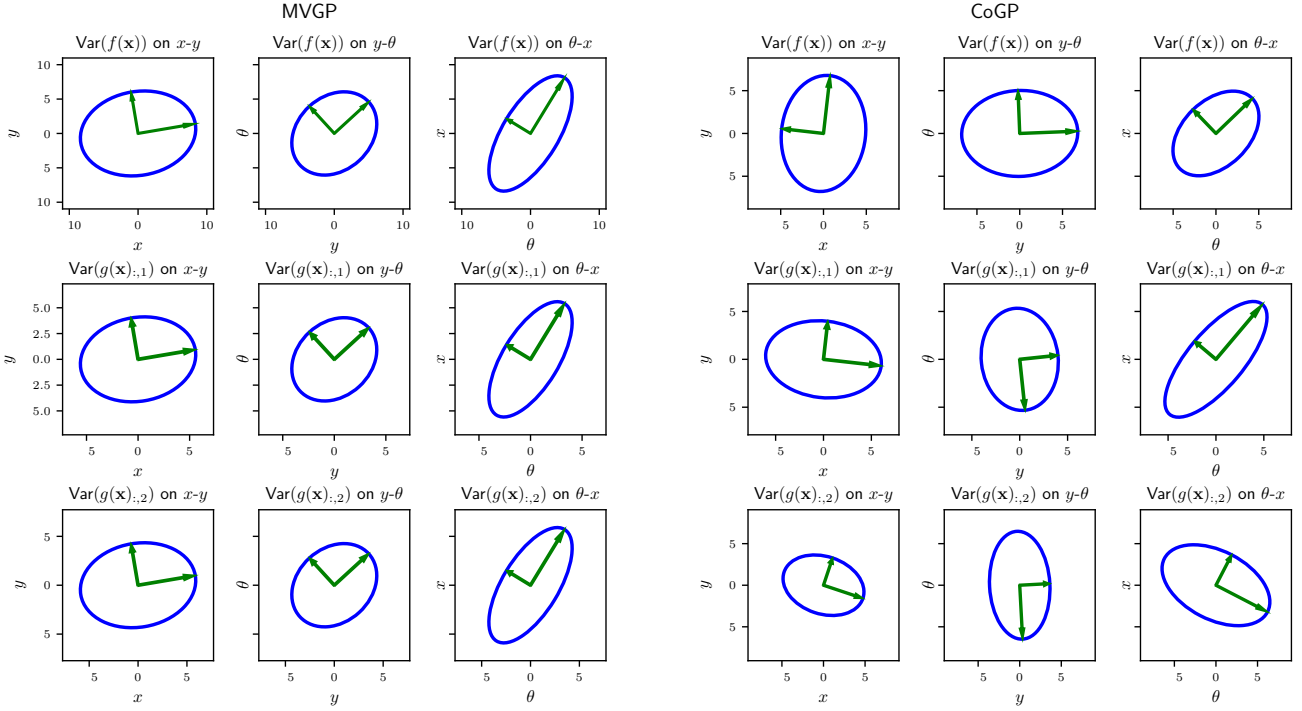


Fig. 1. Covariances of $f(\mathbf{x})$ and the columns of $g(\mathbf{x})$ learned from an Ackermann vehicle simulation described in Sec. VIII-B using an MVGP model (left) and a CoGP model (right). By construction the shapes of the covariances for each column of the MVGP model are the same (up to scale), while the column covariance shapes for the CoGP model are not constrained.

Our approach requires the stochastic system dynamics $F(\mathbf{x})\mathbf{u}$ to be locally Lipschitz continuous with high probability. This is necessary to ensure that for any MVGP sample $\hat{F}(\mathbf{x})$, the ordinary differential equation $\dot{\mathbf{x}} = \hat{F}(\mathbf{x})\mathbf{u}$ in (1) has a unique solution [68, Thm. 3.20]. Without such an assumption, the system would be able to instantaneously move from a safe state with a large safety margin to an unsafe state, making it impossible to guarantee safety for any time duration after a control triggering instant. The next Lemma shows constructively how to compute a Lipschitz constant L_F which holds with a user-specified probability $1 - \delta_L$ for a vector-valued GP model with kernel whose fourth-order partial derivatives are Lipschitz continuous. The proof follows, with minor modifications, the argument in [69, Thm 3.2].

Lemma 1. Consider the vector-valued function $\mathbf{f}(\mathbf{x}) \triangleq F(\mathbf{x})\mathbf{u}$ in (11) with distribution $\mathcal{GP}(\boldsymbol{\mu}(\mathbf{x}), \mathbf{A}\kappa_{\mathbf{B}}(\mathbf{x}, \mathbf{x}'))$, where $\boldsymbol{\mu}(\mathbf{x}) = \mathbf{M}_k(\mathbf{x})\mathbf{u}$ and $\kappa_{\mathbf{B}}(\mathbf{x}, \mathbf{x}') = \mathbf{u}^\top \mathbf{B}_k(\mathbf{x}, \mathbf{x}')\mathbf{u}$. Suppose that $\kappa_{\mathbf{B}}(\mathbf{x}, \mathbf{x}')$ has continuous partial derivatives of fourth order. For $j \in \{1, \dots, n\}$, let $\kappa_{\mathbf{B}}^{\partial_j}(\mathbf{x}, \mathbf{x}') \triangleq \frac{\partial^2}{\partial \mathbf{x}_j \partial \mathbf{x}'_j} \kappa_{\mathbf{B}}(\mathbf{x}, \mathbf{x}')$ with Lipschitz constant $L_{\kappa}^{\partial_j}$ over a compact set \mathcal{X} with maximal extension $r = \max_{\mathbf{x}, \mathbf{x}' \in \mathcal{X}} \|\mathbf{x} - \mathbf{x}'\|$. Then, each element of the Jacobian of $\mathbf{f}(\mathbf{x})$ is bounded with probability of at least $1 - \frac{\delta_L}{n^2}$:

$$\left| \sup_{\mathbf{x} \in \mathcal{X}} \frac{\partial \mathbf{f}_i(\mathbf{x})}{\partial \mathbf{x}_j} \right| \leq L_{i, \partial_j}, \quad \forall i, j \in \{1, \dots, n\}, \quad (12)$$

where

$$L_{i, \partial_j} \triangleq \sqrt{2 \log \left(\frac{2n^2}{\delta_L} \right)} \kappa_{\mathcal{X}}^{i, \partial_j} + 12\sqrt{6n} \max \left\{ \kappa_{\mathcal{X}}^{i, \partial_j}, \sqrt{r \mathbf{A}_{ii} L_{\kappa}^{\partial_j}} \right\}$$

and $\kappa_{\mathcal{X}}^{i, \partial_j} \triangleq \max_{\mathbf{x} \in \mathcal{X}} \sqrt{\mathbf{A}_{ii} \kappa_{\mathbf{B}}^{\partial_j}(\mathbf{x}, \mathbf{x}')}$. Furthermore, a sample function $\mathbf{f}(\mathbf{x})$ of the given GP is almost surely continuous on \mathcal{X} and with probability of at least $1 - \delta_L$, $L_F \triangleq \sqrt{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n L_{i, \partial_j}^2}$ is a Lipschitz constant of $\mathbf{f}(\mathbf{x})$ on \mathcal{X} .

Proof. See Appendix D. \square

The assumptions of Lemma 1 are mild and reasonable. The assumption about kernel continuity can be satisfied by an appropriate choice of a kernel function. For example, commonly used radial basis functions are infinitely differentiable and continuous. The assumption that the space \mathcal{X} is bounded is reasonable for physical systems. For example, the region that an Ackermann vehicle can cover can be bounded based on the maximum velocity and acceleration of the vehicle.

A. Probabilistic Safety Constraints

Consider probabilistic stability and safety constraints in the CLF-CBF QP in (4), induced by the MVGP distribution of $F(\mathbf{x})\mathbf{u}$ at time t_k in (11):

$$\begin{aligned} \pi(\mathbf{x}_k) \in \arg \min_{\mathbf{u}_k \in \mathcal{U}, \delta \in \mathbb{R}} \|\mathbf{R}(\mathbf{x}_k)\mathbf{u}_k\|^2 + \lambda \delta^2 \\ \text{s.t. } \mathbb{P}(\text{CLC}(\mathbf{x}_k, \mathbf{u}_k) \leq \delta \mid \mathbf{x}_k, \mathbf{u}_k) \geq \tilde{p}_k \\ \mathbb{P}(\text{CBC}(\mathbf{x}_k, \mathbf{u}_k) \leq \zeta \mid \mathbf{x}_k, \mathbf{u}_k) \geq \tilde{p}_k, \end{aligned} \quad (13)$$

where $\tilde{p}_k \triangleq p_k / (1 - \delta_L)$ and δ_L is specified in Lemma 1.

Remark 1. Given a desired safety probability p_k for the interval $[t_k, t_k + \tau_k)$, defined in (6), we choose the Lipschitz continuity probability $1 - \delta_L$ in Lemma 1 to ensure instantaneous safety at time t_k with higher probability $\tilde{p}_k \in [0.5, 1)$. The choice of δ_L affects the Lipschitz constant L_F and in turn

the duration τ_k , as discussed in Sec. V-B. By ensuring safety at time instant t_k with probability at least \tilde{p}_k , we guarantee safety with probability at least $p_k = \tilde{p}_k(1 - \delta_L)$ during the interval $[t_k, t_k + \tau_k)$ (see Proposition 5). Also see Remark 2 for further discussion on feasibility. •

To ensure that the safety constraint does not only hold instantaneously at t_k but over an interval $[t_k, t_k + \tau_k)$, we enforce a tighter constraint via $\zeta > 0$ and determine the time τ_k for which it remains valid. The choice of ζ and its effect on τ_k are discussed next. First, we obtain the mean and variance of the CBC constraint, which is an affine transformation of the system dynamics.

Lemma 2. Consider the dynamics in (1) with posterior distribution in (11). Given \mathbf{x}_k and \mathbf{u}_k , $\text{CBC}_k \triangleq \text{CBC}(\mathbf{x}_k, \mathbf{u}_k)$ is a Gaussian random variable with the following parameters:

$$\mathbb{E}[\text{CBC}_k] = \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \mathbf{M}_k(\mathbf{x}_k) \mathbf{u}_k + \alpha(h(\mathbf{x}_k)), \quad (14)$$

$$\text{Var}[\text{CBC}_k] = (\mathbf{u}_k^\top \mathbf{B}_k(\mathbf{x}_k, \mathbf{x}_k) \mathbf{u}_k) (\nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \mathbf{A} \nabla_{\mathbf{x}} h(\mathbf{x}_k)).$$

Proof. We start by rewriting the definition of CBC as:

$$\text{CBC}_k = \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top F(\mathbf{x}_k) \mathbf{u}_k + \alpha(h(\mathbf{x}_k)). \quad (15)$$

Then, the mean and variance of CBC_k are:

$$\mathbb{E}[\text{CBC}_k] = \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \mathbb{E}[F(\mathbf{x}_k) \mathbf{u}_k] + \alpha(h(\mathbf{x}_k)), \quad (16)$$

$$\text{Var}[\text{CBC}_k] = \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \text{Var}[F(\mathbf{x}_k) \mathbf{u}_k] \nabla_{\mathbf{x}} h(\mathbf{x}_k),$$

where the mean and variance of $F(\mathbf{x}_k) \mathbf{u}_k$ are from (11). □

Using Lemma 2, we can rewrite the safety constraint as

$$\mathbb{P}(\text{CBC}_k \geq \zeta | \mathbf{x}_k, \mathbf{u}_k) = 1 - \Phi\left(\frac{\zeta - \mathbb{E}[\text{CBC}_k]}{\sqrt{\text{Var}[\text{CBC}_k]}}\right) \geq \tilde{p}_k, \quad (17)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian. Note that if the control input is chosen so that $\frac{\zeta - \mathbb{E}[\text{CBC}_k]}{\sqrt{\text{Var}[\text{CBC}_k]}} \rightarrow -\infty$, as the posterior variance of CBC_k tends to zero, the probability $\mathbb{P}(\text{CBC}_k \geq \zeta | \mathbf{x}_k, \mathbf{u}_k)$ tends to one. Namely, as the uncertainty about the system dynamics tends to zero, our results reduce to the setting of Sec. III-B, and safety can be ensured with probability one. Using (17) and noting that $\Phi^{-1}(1 - \tilde{p}_k) = -\sqrt{2} \text{erf}^{-1}(2\tilde{p}_k - 1)$, where $\text{erf}(\cdot)$ is the Gauss error function, the optimization problem in (13) can be restated as:

$$\begin{aligned} \pi(\mathbf{x}_k) \in \arg \min_{\mathbf{u}_k \in \mathcal{U}, \delta \in \mathbb{R}} \quad & \|\mathbf{R}(\mathbf{x}_k) \mathbf{u}_k\|^2 + \lambda \delta^2 \\ \text{s.t.} \quad & \delta - \mathbb{E}[\text{CLC}_k] \geq c(\tilde{p}_k) \sqrt{\text{Var}[\text{CLC}_k]} \\ & \zeta - \mathbb{E}[\text{CBC}_k] \leq -c(\tilde{p}_k) \sqrt{\text{Var}[\text{CBC}_k]}, \end{aligned} \quad (18)$$

where $c(\tilde{p}_k) \triangleq \sqrt{2} \text{erf}^{-1}(2\tilde{p}_k - 1)$. Using the reformulation in (18), the next proposition shows that the control problem with probabilistic stability and safety constraints in (13) is a convex optimization problem, which can be solved efficiently.

Proposition 4. The chance-constrained optimization problem in (13) for control-affine system dynamics in (1) with posterior distribution (11) is a second-order cone program (SOCP):

$$\pi(\mathbf{x}_k) \in \arg \min_{y \in \mathbb{R}, \delta \in \mathbb{R}, \mathbf{u} \in \mathcal{U}} \quad y \quad (19a)$$

$$\text{s.t.} \quad y - \left\| \mathbf{R}(\mathbf{x}_k) \mathbf{u} + \sqrt{\lambda} \delta \right\|_2 \geq 0 \quad (19b)$$

$$\mathbf{q}_k(\mathbf{x}_k)^\top \mathbf{u} - \delta + c(\tilde{p}_k) \|\mathbf{P}_k(\mathbf{x}_k) \mathbf{u}\|_2 \leq 0 \quad (19c)$$

$$\mathbf{e}_k(\mathbf{x}_k)^\top \mathbf{u} - \zeta - c(\tilde{p}_k) \|\mathbf{V}_k(\mathbf{x}_k) \mathbf{u}\|_2 \geq 0, \quad (19d)$$

where

$$\mathbf{q}_k(\mathbf{x}_k) \triangleq \mathbf{M}_k(\mathbf{x}_k)^\top \nabla_{\mathbf{x}} V(\mathbf{x}_k) + \begin{bmatrix} \gamma(V(\mathbf{x}_k)) \\ \mathbf{0}_m \end{bmatrix},$$

$$\mathbf{P}_k(\mathbf{x}_k) \triangleq \sqrt{\nabla_{\mathbf{x}} V(\mathbf{x}_k)^\top \mathbf{A} \nabla_{\mathbf{x}} V(\mathbf{x}_k)} \mathbf{B}_k^{\frac{1}{2}}(\mathbf{x}_k, \mathbf{x}_k),$$

$$\mathbf{e}_k(\mathbf{x}_k) \triangleq \mathbf{M}_k(\mathbf{x}_k)^\top \nabla_{\mathbf{x}} h(\mathbf{x}_k) + \begin{bmatrix} \alpha(h(\mathbf{x}_k)) \\ \mathbf{0}_m \end{bmatrix},$$

$$\mathbf{V}_k(\mathbf{x}_k) \triangleq \sqrt{\nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \mathbf{A} \nabla_{\mathbf{x}} h(\mathbf{x}_k)} \mathbf{B}_k^{\frac{1}{2}}(\mathbf{x}_k, \mathbf{x}_k),$$

and $\mathbf{B}_k^{\frac{1}{2}}(\mathbf{x}_k, \mathbf{x}_k)$ is the Cholesky factorization of $\mathbf{B}_k(\mathbf{x}_k, \mathbf{x}_k)$.

Proof. Substituting the mean and variance of CLC_k and CBC_k (Lemma 2) in (18) and introducing an auxiliary variable y to express the quadratic objective as a SOCP constraint leads to the result. □

Prop. 4 generalizes the CLF-CBF QP formulation in (4). It shows that when the stability and safety constraints are probabilistic, induced by the MVGP distribution of the system dynamics, the control synthesis problem becomes a SOCP and, hence, is still convex. We refer to the left-hand side of the constraints in (19d) and (19c) as Stochastic CBC (SCBC) and Stochastic CLC (SCLC), respectively. Note that when $\tilde{p}_k = 0.5$, $c(\tilde{p}_k) = 0$ and the SOCP in (19) reduces to the deterministic-case QP in (4). Otherwise, the SOCP can be solved to arbitrary precision ϵ within $O(-\log(\epsilon))$ iterations [70] with off-the-shelf solvers [71], [72].

Remark 2. If the SOCP program in (19) is infeasible, there does not exist a control input that guarantees safety for the current dynamics GP distribution and the user-specified probabilistic safety requirement. This can be used as a failure mode to ask a human to take over control or provide additional training data that reduces the dynamics GP uncertainty. Since our SOCP constraint is a conservative approximation of the true probabilistic safety constraint in (13), it is possible that control that meets the desired safety bounds exists even if the SOCP is infeasible. Please refer to Casteñada et al [73] for discussion of feasibility conditions of an SOCP controller. •

B. Self-triggering Design

The safety constraint in (19) ensures safety with high probability at the triggering times $\{t_k\}_{k \in \mathbb{N}}$. Here, we extend our analysis to the inter-triggering intervals $[t_k, t_k + \tau_k)$. We restate [30, Prop. 1] in our setting, showing that when the system dynamics are Lipschitz continuous (Lemma 1), then the change in the state is bounded.

Lemma 3. Consider the GP model of the system dynamics in (11) with zero-order hold control for $[t_k, t_k + \tau_k)$. If the assumptions of Lemma 1 are satisfied, i.e., with high probability the system dynamics are Lipschitz continuous with Lipschitz constant $L_{\mathbf{f}_k}$, then for all $s \in [0, \tau_k)$ the total change in the system state is bounded with probability at least $1 - \delta_L$:

$$\|\mathbf{x}(t_k + s) - \mathbf{x}_k\| \leq \bar{r}_k(s) \triangleq \frac{1}{L_{\mathbf{f}_k}} \|\dot{\mathbf{x}}_k\| (e^{sL_{\mathbf{f}_k}} - 1). \quad (20)$$

Recall from Sec. III-B that h is a continuously differentiable function. Thus, using Lemma 3, for any inter-triggering time τ_k , there exist a constant $L_{h_k} > 0$ such that:

$$\sup_{s \in [0, \tau_k]} \|\nabla_{\mathbf{x}} h(\mathbf{x}(t_k + s))\| \leq L_{h_k}. \quad (21)$$

This is used in the next proposition which concerns Problem 2. The next proposition guarantees safety during the inter-triggering intervals as long as the controller ensures safety at the triggering time using the SOCP in (19).

Proposition 5. *Consider the GP model of the system dynamics in (11) with safe set \mathcal{C} . Assume the system dynamics have Lipschitz constant L_{f_k} with probability at least $1 - \delta_L$ (Lemma 1), the system is safe with margin ζ at triggering time t_k with probability at least $\tilde{p}_k = p_k / (1 - \delta_L)$ (Prop. 4), and α is Lipschitz continuous with Lipschitz constant L_α . Then, the system is safe according to the constraint in (6) with probability at least p_k for time duration $\tau_k \leq \frac{1}{L_{f_k}} \ln \left(1 + \frac{L_{f_k} \zeta}{(L_{f_k} + L_\alpha) L_{h_k} \|\dot{\mathbf{x}}_k\|} \right)$, where L_{h_k} is defined in (21).*

Proof. Since the program (13) has a solution at the triggering time t_k , we know $\mathbb{P}(\text{CBC}_k \geq \zeta | \mathbf{x}_k, \mathbf{u}_k) \geq \tilde{p}_k$. Thus, conditioned on $\text{CBC}_k \geq \zeta$ and the Lipschitz continuity of the system dynamics, if we prove $\text{CBC}(s + t_k) \geq 0$, for all $s \in [0, \tau_k]$, the result follows. Using the Cauchy-Schwarz inequality, Lipschitz continuity of the system dynamics and Lipschitz continuity of α and h (from (21)), for all $s \in [0, \tau_k]$, we have with probability at least $\tilde{p}_k(1 - \delta_L)$:

$$\begin{aligned} & |\text{CBC}(\mathbf{x}(s + t_k), \mathbf{u}_k) - \text{CBC}_k| \\ & \leq \left(\sup_{s \in [0, \tau_k]} \|\nabla_{\mathbf{x}} h(\mathbf{x}(s + t_k))\| L_{f_k} + L_\alpha L_{h_k} \right) \bar{\tau}_k(s). \end{aligned} \quad (22)$$

Thus, using (20) and (21), we notice that the right-hand side of (22) is upper-bounded by $\frac{L_{h_k} L_{f_k} + L_\alpha L_{h_k}}{L_{f_k}} \|\dot{\mathbf{x}}_k\| (e^{L_{f_k} s} - 1)$, which, in turn, is less than or equal to ζ for $s = \tau_k$. \square

Prop. 4 ensures instantaneous safety with probability $\tilde{p}_k \geq p_k$, while Prop. 5 extends the safety guarantee with probability p_k to the interval $[t_k, t_k + \tau_k)$. System safety as formalized in Problem 2 is guaranteed because $\mathbb{P}(\text{CBC}(\mathbf{x}) \geq 0) \geq p_k$ implies $\mathbb{P}(h(\mathbf{x}) \geq 0) \geq p_k$, for all $t \in [t_k, t_k + \tau_k)$, due to Prop. 2. Prop. 5 characterizes the longest time τ_k until it is necessary to recompute the control input to guarantee safety.

Fig. 2 shows an evaluation of the Lipschitz constant L_{f_k} (Lemma 1) and triggering time τ_k (Prop. 5) for a system with Ackermann dynamics. We compute L_{f_k} and τ_k both analytically and numerically. The shapes of the curves for the numerical and analytical Lipschitz constant are the same, although the analytical bound in Lemma 1 is a few orders of magnitude more conservative. More details about the simulation are presented in Sec. VIII-B.

VI. EXTENSION TO HIGHER RELATIVE-DEGREE SYSTEMS

Next, we extend the probabilistic safety constraint formulation for systems with arbitrary relative degree², using

²The motivation for assuming known relative degree but unknown dynamics comes from robotics applications. Commonly, the class of the system is known but the parameters (e.g., mass, moment of inertia) and high-order interactions (e.g., jerk, snap, drag) of the dynamics are unknown. Estimating the relative degree is left for future work (see [74]–[76]).

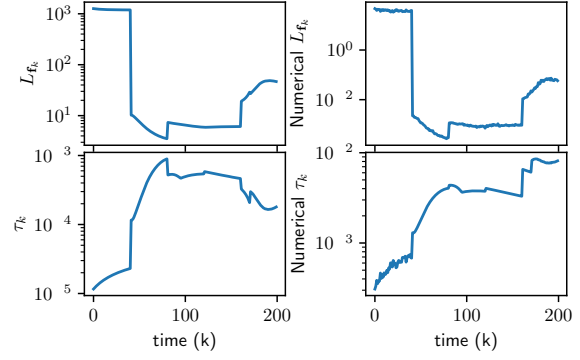


Fig. 2. Lipschitz constant L_{f_k} and triggering time τ_k for an Ackermann vehicle simulation. The analytical values (left) obtained from Lemma 1 and Prop. 5 are compared with numerical estimates (right). The shapes are similar for both terms over a 200 time-step simulation but the magnitudes differs. See Sec. VIII-B3 for details.

an exponential control barrier function (ECBF) [25], [37]. Sec. VI-A reviews ECBF results for systems with *known dynamics*. Sec. VI-B investigates probabilistic ECBF constraints, while Sec. VI-C provides a self-triggering formulation.

A. Known Dynamics

Let $r \geq 1$ be the relative degree of (1) with respect to $h(\mathbf{x})$, i.e., $\mathcal{L}_g \mathcal{L}_f^{(r-1)} h(\mathbf{x}) \neq 0$ and $\mathcal{L}_g \mathcal{L}_f^{(k-1)} h(\mathbf{x}) = 0$, $\forall k \in \{1, \dots, r-1\}$. This condition can be expressed by defining

$$\boldsymbol{\eta}(\mathbf{x}) \triangleq \begin{bmatrix} h(\mathbf{x}) \\ \mathcal{L}_f h(\mathbf{x}) \\ \vdots \\ \mathcal{L}_f^{(r-1)} h(\mathbf{x}) \end{bmatrix}, \quad \mathcal{F} \triangleq \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathcal{G} \triangleq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

and $h(\mathbf{x})$ is the output of the linear time-invariant system:

$$\dot{\boldsymbol{\eta}}(\mathbf{x}) = \mathcal{F} \boldsymbol{\eta}(\mathbf{x}) + \mathcal{G} \mathbf{u}, \quad h(\mathbf{x}) = \mathbf{c}^\top \boldsymbol{\eta}(\mathbf{x}), \quad (23)$$

where $\mathbf{c} \triangleq [1, 0, \dots, 0]^\top \in \mathbb{R}^r$.

Definition 4 ([25], [37]). A function $h \in \mathbb{C}^r(\mathcal{X}, \mathbb{R})$ is an exponential control barrier function (ECBF) for the system in (1) if there exists a vector $\mathbf{k}_\alpha \in \mathbb{R}^r$ such that the r -th order control barrier condition $\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u}) \triangleq \mathcal{L}_f^{(r)} h(\mathbf{x}) + \mathcal{L}_g \mathcal{L}_f^{(r-1)} h(\mathbf{x}) \mathbf{u} + \mathbf{k}_\alpha^\top \boldsymbol{\eta}(\mathbf{x})$ satisfies $\sup_{\mathbf{u} \in \mathcal{U}} \text{CBC}^{(r)}(\mathbf{x}, \mathbf{u}) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$ and $h(\mathbf{x}(t)) \geq \mathbf{c}^\top \boldsymbol{\eta}(\mathbf{x}_0) e^{(\mathcal{F} - \mathcal{G} \mathbf{k}_\alpha) t} \geq 0$, whenever $h(\mathbf{x}(0)) \geq 0$.

If \mathbf{k}_α satisfies the properties in [37, Thm. 2], then any controller $\mathbf{u} = \pi(\mathbf{x})$ that ensures $\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u}) \geq 0$ renders the dynamics (1) safe with respect to $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$. For relative-degree-one systems, $\mathbf{k}_\alpha^\top \boldsymbol{\eta}(\mathbf{x})$ reduces to $\alpha h(\mathbf{x})$ with $\alpha > 0$. Thus, $\text{CBC}^{(1)}$, the safety condition based on ECBF for $r = 1$, is equivalent to the CBC in Def. 2 when the extended class K_∞ function α is linear.

B. Probabilistic ECBF Constraints

As in (13), we are interested in solving

$$\begin{aligned} \pi(\mathbf{x}_k) \in \arg \min_{\mathbf{u}_k \in \mathcal{U}} & \|\mathbf{R}(\mathbf{x}_k) \mathbf{u}_k\|_2 \\ \text{s.t. } & \mathbb{P}(\text{CBC}_k^{(r)} \geq \zeta | \mathbf{x}_k, \mathbf{u}_k) \geq \tilde{p}_k, \end{aligned} \quad (24)$$

where $\text{CBC}_k^{(r)} \triangleq \text{CBC}^{(r)}(\mathbf{x}_k, \mathbf{u}_k)$ and we have dropped the CLC term for clarity of presentation. While we explicitly characterised the distribution of $\text{CBC}_k^{(r)}$ for relative degree one in Lemma 2, the distribution of $\text{CBC}_k^{(r)}$ cannot be determined explicitly for arbitrary r . Instead, we use a concentration bound to rewrite the chance constraint in terms of the moments of $\text{CBC}_k^{(r)}$. Before we can derive a SOCP controller for higher relative-degree systems, we need to show that the mean and variance of $\text{CBC}_k^{(r)}$ are affine and quadratic in \mathbf{u} , respectively.

Lemma 4. *Let $r \geq 1$ be the relative degree of (1) with respect to $h(\mathbf{x}) \in \mathcal{C}^r(\mathcal{X}, \mathbb{R})$ and $F(\mathbf{x})$ be a random function with finite mean and variance. Then, the expectation and variance of $\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})$ in Def. 4 are linear and quadratic in \mathbf{u} , respectively, and satisfy:*

$$\begin{aligned} \mathbb{E}[\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})] &= \mathbf{e}^{(r)}(\mathbf{x})^\top \mathbf{u}, \\ \text{Var}[\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})] &= \mathbf{u}^\top \mathbf{V}^{(r)}(\mathbf{x}) \mathbf{V}^{(r)}(\mathbf{x})^\top \mathbf{u}, \end{aligned} \quad (25)$$

where

$$\mathbf{e}^{(r)}(\mathbf{x}) \triangleq \mathbb{E} \left[F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f^{(r-1)} h(\mathbf{x}) + \begin{bmatrix} \mathbf{k}_\alpha^\top \boldsymbol{\eta}(\mathbf{x}) \\ \mathbf{0}_m \end{bmatrix} \right] \quad (26)$$

$$\mathbf{V}^{(r)}(\mathbf{x}) \triangleq \text{Var} \left[F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f^{(r-1)} h(\mathbf{x}) + \begin{bmatrix} \mathbf{k}_\alpha^\top \boldsymbol{\eta}(\mathbf{x}) \\ \mathbf{0}_m \end{bmatrix} \right]^\frac{1}{2} \quad (27)$$

Proof. See Appendix E. \square

The next proposition generalizes Prop. 4 to arbitrary relative-degree systems by showing that control synthesis with probabilistic safety constraints can still be cast as a SOCP.

Proposition 6. *Consider the control-affine system in (1) with stochastic dynamics and relative degree $r \geq 1$ with respect to an ECBF $h(\mathbf{x}) \in \mathcal{C}^r(\mathcal{X}, \mathbb{R})$. If the system inputs are determined in continuous time t_k from the following SOCP,*

$$\begin{aligned} \pi(\mathbf{x}_k) &\in \arg \min_{y \in \mathbb{R}, \mathbf{u}_k \in \mathcal{U}} y \\ \text{s.t. } &y - \|\mathbf{R}(\mathbf{x}_k) \mathbf{u}_k\|_2 \geq 0 \\ &\mathbf{e}_k^{(r)}(\mathbf{x}_k)^\top \mathbf{u}_k - \zeta - c^{(r)}(\tilde{p}_k) \left\| \mathbf{V}_k^{(r)}(\mathbf{x}_k) \mathbf{u}_k \right\|_2 \geq 0, \end{aligned} \quad (28)$$

where $c^{(r)}(\tilde{p}_k) \triangleq \sqrt{\frac{\tilde{p}_k}{1-\tilde{p}_k}}$ and $\tilde{p}_k \in (0, 1)$ is fixed for all t_k , then the system trajectory remains in the safe set $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$ with probability \tilde{p}_k .

Proof. See Appendix F. \square

In the proposition above, we bound $\mathbb{P}(\text{CBC}_k^{(r)} \geq \zeta \mid \mathbf{x}_k, \mathbf{u}_k)$ in (24) using Cantelli's inequality, because the exact distribution of $\text{CBC}_k^{(r)}$ is unknown. The safety constraint in (28) can be interpreted as: "The system must satisfy $\text{CBC}_k^{(r)}$ in expectation by a margin $c(\tilde{p}_k)$ times the standard deviation of $\text{CBC}_k^{(r)}$ ". Solving the SOCP requires knowledge of the mean and variance of $\text{CBC}_k^{(r)}$. Sec. VII shows how to determine the mean and variance of $\text{CBC}_k^{(2)}$, which is of practical relevance for many physical systems. For $r > 2$, Monte Carlo sampling can be used to estimate these quantities.

C. Self-triggering Design

In this section, we extend the time-triggered formulation for probabilistic safety of high relative-degree systems to a self-triggering setup. Our extension to self-triggering control in the relative-degree-one case in Proposition 5, relied on Lipschitz continuity of the CBC components. To simplify the analysis for arbitrary relative degree r , we temporarily introduce a modified CBF $h_b(\mathbf{x}) \triangleq h(\mathbf{x}) - \zeta_b$ with $\zeta_b > 0$. We solve the SOCP in (28) with $\zeta = 0$ but replace $h(\mathbf{x})$ with $h_b(\mathbf{x})$. Enforcing the safety constraint in (28) for $h_b(\mathbf{x}_k)$, ensures that with probability \tilde{p}_k , $h(\mathbf{x}_k) \geq \zeta_b$ at the sampling times. We find an upper bound on the sampling time τ_k that ensures $h(\mathbf{x}(t))$ remains non-negative during the inter-triggering intervals $[t_k, t_k + \tau_k)$. As shown in the next proposition, this is sufficient to guarantee safety during the inter-triggering interval as long as the system using the SOCP controller (28) is safe at the triggering time t_k .

Proposition 7. *Consider the system in (1) with stochastic dynamics and safe set $\mathcal{C} = \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$. Suppose the system has relative degree $r \geq 1$ with respect to $h(\mathbf{x})$ and the SOCP in (28) has a solution at triggering time t_k with $h_b(\mathbf{x}) \triangleq h(\mathbf{x}) - \zeta_b$ as the CBF and $\zeta = 0$. Suppose that the system dynamics are Lipschitz continuous with probability at least $1 - \delta_L$ (Lemma 1) and h is Lipschitz continuous with Lipschitz constant L_{h_k} . Then, $h(\mathbf{x}(t)) \geq 0$, for $t \in [t_k, t_k + \tau_k)$, holds with probability $p_k = \tilde{p}_k(1 - \delta_L)$, where $\tau_k \leq \frac{1}{L_{f_k}} \ln \left(1 + \frac{L_{f_k} \zeta_b}{L_{h_k} \|\dot{\mathbf{x}}_k\|} \right)$.*

Proof. Using Lemma 1 and 3 we have,

$$\begin{aligned} \sup_{s \in [0, \tau_k)} |h(\mathbf{x}(t_k + s)) - h(\mathbf{x}_k)| &\leq L_{h_k} \|\mathbf{x}(t_k + s) - \mathbf{x}_k\| \\ &\leq \frac{L_{h_k}}{L_{f_k}} \|\dot{\mathbf{x}}_k\| (e^{L_{f_k} \tau_k} - 1) \leq \zeta_b, \end{aligned} \quad (29)$$

where the last inequality follows from the upper bound on τ_k . Given $h(\mathbf{x}(t_k)) \geq \zeta_b$ at t_k , we deduce $h(\mathbf{x}(t)) \geq 0$ for all time in $[t_k, t_k + \tau_k)$, and the result follows. \square

Assuming that the Lipschitz continuity of system dynamics occurs with probability at least $1 - \delta_L$ (Lemma 1), and given the value of $\|\dot{\mathbf{x}}_k\|$ at triggering time t_k and the parameter $\zeta_b > 0$ and the Lipschitz constant L_{h_k} , Proposition 7 characterizes the longest time τ_k until a control input needs to be recomputed to guarantee safety with probability at least $p_k = \tilde{p}_k(1 - \delta)$ for a system with arbitrary relative degree.

Remark 3. Recent works [38], [77], [78] study a more general high-order CBF (HOCBF) formulation than ECBF. Applying our probabilistic safety constraints with a HOCBF is left for future work. \bullet

VII. SPECIAL CASE: RELATIVE DEGREE TWO

Systems with relative degree two appear frequently. We develop an efficient approach for computing the mean and variance of the relative-degree-two safety condition:

$$\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u}) = [\nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x})]^\top F(\mathbf{x}) \mathbf{u} + [h(\mathbf{x}), \mathcal{L}_f h(\mathbf{x})]^\top \mathbf{k}_\alpha,$$

which is needed to specify the safety constraints in our SOCP formulation in (28). Note that $\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})$ is a stochastic process whose distribution is induced by the GP $\text{vec}(F(\mathbf{x}))$. Fig. 3

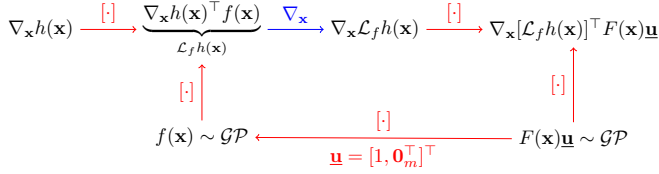


Fig. 3. Computation graph for the second-order Lie derivative of a control barrier function $h(\mathbf{x})$ along the GP-distributed dynamics $F(\mathbf{x})\mathbf{u}$ of a relative degree two system. Only two kinds of operations are required: (1) dot product with a vector-variate GP ([:]) and (2) gradient of a scalar GP ($\nabla_{\mathbf{x}}$).

shows a computation graph for $\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})$ with random or deterministic functions as nodes and computations among them as edges. The graph contains only two kinds of edges: (1) dot product of a deterministic function with a vector-variate GP and (2) gradient of a scalar GP. We show how the mean, variance, and covariance propagate through these two computations in Lemma 5 (dot product) and Lemma 6 (scalar GP gradient).

Lemma 5. Consider three Gaussian random vectors \mathbf{x} , \mathbf{y} , \mathbf{z} with means $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$, $\bar{\mathbf{z}}$ and variances $\text{Var}[\mathbf{x}]$, $\text{Var}[\mathbf{y}]$, $\text{Var}[\mathbf{z}]$, respectively. Let their pairwise covariances be $\text{cov}(\mathbf{x}, \mathbf{y})$, $\text{cov}(\mathbf{y}, \mathbf{z})$ and $\text{cov}(\mathbf{z}, \mathbf{x})$. The mean, variance, and covariances of $\mathbf{x}^\top \mathbf{y}$ are given by:

$$\mathbb{E}[\mathbf{x}^\top \mathbf{y}] = \bar{\mathbf{x}}^\top \bar{\mathbf{y}} + \frac{1}{2} \text{tr}(\text{cov}(\mathbf{x}, \mathbf{y}) + \text{cov}(\mathbf{y}, \mathbf{x})) \quad (30)$$

$$\text{Var}[\mathbf{x}^\top \mathbf{y}] = \frac{1}{2} \text{tr}(\text{cov}(\mathbf{x}, \mathbf{y}) + \text{cov}(\mathbf{y}, \mathbf{x}))^2 + \bar{\mathbf{y}}^\top \text{Var}[\mathbf{x}]\bar{\mathbf{y}} + \bar{\mathbf{x}}^\top \text{Var}[\mathbf{y}]\bar{\mathbf{x}} + \bar{\mathbf{y}}^\top \text{cov}(\mathbf{x}, \mathbf{y})\bar{\mathbf{x}} + \bar{\mathbf{x}}^\top \text{cov}(\mathbf{y}, \mathbf{x})\bar{\mathbf{y}} \quad (31)$$

$$\begin{bmatrix} \text{cov}(\mathbf{x}, \mathbf{x}^\top \mathbf{y}) \\ \text{cov}(\mathbf{y}, \mathbf{x}^\top \mathbf{y}) \\ \text{cov}(\mathbf{z}, \mathbf{x}^\top \mathbf{y}) \end{bmatrix} = \begin{bmatrix} \text{cov}(\mathbf{x}, \mathbf{y})\bar{\mathbf{x}} + \text{Var}[\mathbf{x}]\bar{\mathbf{y}} \\ \text{Var}[\mathbf{y}]\bar{\mathbf{x}} + \text{cov}(\mathbf{y}, \mathbf{x})\bar{\mathbf{y}} \\ \text{cov}(\mathbf{z}, \mathbf{y})\bar{\mathbf{x}} + \text{cov}(\mathbf{z}, \mathbf{x})\bar{\mathbf{y}} \end{bmatrix}. \quad (32)$$

Proof. See Appendix G. \square

Lemma 6 ([79, Thm. 2.2.2][80, p. 43]). Let $q(\mathbf{x})$ be a scalar Gaussian Process with differentiable mean function $\mu(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ and twice-differentiable covariance function $\kappa(\mathbf{x}, \mathbf{x}') : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$. If $\nabla_{\mathbf{x}}\mu(\mathbf{x})$ exists and is finite for all $\mathbf{x} \in \mathbb{R}^n$ and $\mathcal{H}_{\mathbf{x}, \mathbf{x}'}\kappa(\mathbf{x}, \mathbf{x}') = [\frac{\partial^2 \kappa(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x'_j}]_{i=1, j=1}^{n, n}$ exists and is finite for all $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{2n}$, then $q(\mathbf{x})$ possesses a mean square derivative $\nabla_{\mathbf{x}}q(\mathbf{x})$, which is a vector-variate Gaussian Process $\mathcal{GP}(\nabla_{\mathbf{x}}\mu(\mathbf{x}), \mathcal{H}_{\mathbf{x}, \mathbf{x}'}\kappa(\mathbf{x}, \mathbf{x}'))$. If s is another random process whose finite covariance with q is $\text{cov}_{q,s}(\mathbf{x}, \mathbf{x}')$, then

$$\text{cov}(\nabla_{\mathbf{x}}q(\mathbf{x}), s(\mathbf{x}')) = \nabla_{\mathbf{x}} \text{cov}_{q,s}(\mathbf{x}, \mathbf{x}'). \quad (33)$$

The mean and variance of $\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})$ can be computed using Lemma 4:

$$\begin{aligned} \mathbb{E}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})] &= \mathbf{e}^{(2)}(\mathbf{x})^\top \mathbf{u} \\ \text{Var}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})] &= \mathbf{u}^\top \mathbf{V}^{(2)}(\mathbf{x}) \mathbf{V}^{(2)}(\mathbf{x})^\top \mathbf{u} \\ \mathbf{e}^{(2)}(\mathbf{x}) &= \mathbb{E}[F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x})] \\ &\quad + \begin{bmatrix} \mathbf{k}_{\alpha,1} h(\mathbf{x}) + \mathbf{k}_{\alpha,2} \mathbb{E}[\mathcal{L}_f h(\mathbf{x})] \\ \mathbf{0}_m \end{bmatrix} \\ \mathbf{V}^{(2)}(\mathbf{x}) &= \left(\text{Var}[F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x})] \right. \end{aligned} \quad (34)$$

Algorithm 1: Mean and variance of $\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})$.

Data: ECBF $h(\mathbf{x})$, system dynamics distribution

$$\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{B}_k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A})$$

Result: $\mathbb{E}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})]$ and $\text{Var}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})]$

- 1 Compute $\mathbb{E}[\mathcal{L}_f h(\mathbf{x})] = \nabla_{\mathbf{x}} h(\mathbf{x})^\top \mathbf{M}_k(\mathbf{x}) [1, \mathbf{0}_m^\top]^\top$ and $\text{Var}[\mathcal{L}_f h(\mathbf{x})] = [\mathbf{B}_k(\mathbf{x}, \mathbf{x}')]_{1,1} (\nabla_{\mathbf{x}} h(\mathbf{x})^\top \mathbf{A} \nabla_{\mathbf{x}} h(\mathbf{x}))$ by substituting $\mathbf{u} = [1, \mathbf{0}_m^\top]^\top$ in Lemma 2.
 - 2 Compute the mean and variance of $\nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x})$ using Lemma 6,
 - 3 Compute $\text{cov}(\nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x}), \mathcal{L}_f h(\mathbf{x}))$ using (33)
 - 4 Compute the mean and variance $\nabla_{\mathbf{x}} [\mathcal{L}_f h(\mathbf{x})]^\top F(\mathbf{x})\mathbf{u}$ using Lemma 5,
 - 5 Compute $\mathbf{d}(\mathbf{x})^\top \mathbf{u} = \text{cov}(\nabla_{\mathbf{x}} [\mathcal{L}_f h(\mathbf{x})]^\top F(\mathbf{x})\mathbf{u}, \mathcal{L}_f h(\mathbf{x}))$ using (32),
 - 6 Plug the above values into (34) to obtain $\mathbb{E}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})]$ and $\text{Var}[\text{CBC}^{(2)}(\mathbf{x}, \mathbf{u})]$.
-

$$+ \begin{bmatrix} \mathbf{k}_{\alpha,2}^2 \text{Var}[\mathcal{L}_f h(\mathbf{x})] + 2[\mathbf{d}(\mathbf{x})]_1 & [\mathbf{d}(\mathbf{x})]_{2:m+1}^\top \\ [\mathbf{d}(\mathbf{x})]_{2:m+1} & \mathbf{0}_{m \times m} \end{bmatrix}^{\frac{1}{2}}$$

$$\mathbf{d}(\mathbf{x}) \triangleq \mathbf{k}_{\alpha,2} \text{cov}(F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x}), \mathcal{L}_f h(\mathbf{x}))$$

where the means, variances, and covariances of $\mathcal{L}_f h(\mathbf{x})$ and $F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f h(\mathbf{x})$ in the above equations can be computed via Algorithm 1 using Lemma 5 and Lemma 6.

VIII. SIMULATIONS

We evaluate our proposed MVGP learning model and the SOCP-based safe control synthesis on two simulated systems: (A) Pendulum and (B) Ackermann Vehicle. To allow reproducing the results, our implementation is available on Github³.

A. Pendulum

Consider a pendulum, shown in Fig 6, with state $\mathbf{x} = [\theta, \omega]^\top$, containing its angular deviation θ from the rest position and its angular velocity ω . The pendulum dynamics are:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} \omega \\ -\frac{g}{l} \sin(\theta) \end{bmatrix}}_{f(\mathbf{x})} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{ml} \end{bmatrix}}_{g(\mathbf{x})} u, \quad (35)$$

where g is the gravitational acceleration, m is the mass, and l is the length.

1) *Estimating Pendulum Dynamics:* We compare our MVGP model versus the Coregionalization GP (CoGP) [65] in estimating the pendulum dynamics using data from randomly generated control inputs. In our simulation, the true pendulum model has mass $m = 1$ and length $l = 1$. The inference algorithms were implemented in Python using GPyTorch [81]. We let $\mathbf{K}_0(\mathbf{x}, \mathbf{x}') = \mathbf{B}\kappa_0(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A}$ for MVGP and $\mathbf{K}_0(\mathbf{x}, \mathbf{x}') = \Sigma\kappa_0(\mathbf{x}, \mathbf{x}')$ for CoGP, where $\kappa_0(\mathbf{x}, \mathbf{x}')$ is a scaled radial-basis function kernel. Further, \mathbf{A} , \mathbf{B} , and Σ are constrained to be positive semi-definite by modeling each as $\mathbf{C}_r \mathbf{C}_r^\top + \text{diag}(\mathbf{v})$, where $\mathbf{C}_r \in \mathbb{R}^{l \times r}$, r is a desired rank and $\text{diag}(\mathbf{v})$ is a diagonal matrix constructed from a vector \mathbf{v} .

³https://github.com/wecacuee/Bayesian_CBF

We emphasize that a straightforward implementation of decoupled GPs for each system dimension is not possible because the training data is only available as a linear combination of the unknown functions $f(\mathbf{x})$ and $g(\mathbf{x})$. We approximate decoupled GP inference by constraining the matrices \mathbf{A} , \mathbf{B} , and Σ in the MVGP and CoGP models to be diagonal. To compare the accuracy of the GP models, we randomly split the samples from the pendulum simulation into training data and test data. Given a test set \mathcal{T} , we compute *variance-weighted error* as,

$$\text{err}(\mathcal{T}) \triangleq \sqrt{\frac{\sum_{\mathbf{x} \in \mathcal{T}} \|\mathbf{K}_k^{-\frac{1}{2}}(\mathbf{x}, \mathbf{x}) \text{vec}(\mathbf{M}_k(\mathbf{x}) - F(\mathbf{x}))\|_2^2}{|\mathcal{T}|}}. \quad (36)$$

A qualitative comparison of MVGP and CoGP is shown in Fig. 4. A quantitative comparison of the computational efficiency and inference accuracy of the models with *full* and *diagonal* covariance matrices is shown in Fig. 5. The median variance-weighted error and error bars showing the 2nd to 9th decile over 30 repetitions of the evaluation are shown. The experiments are performed on a desktop with Nvidia[®] GeForce RTX[™] 2080Ti GPU and Intel[®] Core[™] i9-7900X CPU (3.30GHz). The results show that MVGP inference is significantly faster than CoGP, while maintaining comparable accuracy. Both MVGP and CoGP have higher median accuracy than their counterparts with diagonally restricted covariances.

2) *Safe Control of Learned Pendulum Dynamics*: A safe set is chosen as the complement of a radial region $[\theta_c - \Delta_{col}, \theta_c + \Delta_{col}]$ with $\theta_c = 45^\circ$, $\Delta_{col} = 22.5^\circ$ that needs to be avoided, as shown in Fig. 6. The control barrier function defining this safe set is $h(\mathbf{x}) = \cos(\Delta_{col}) - \cos(\theta - \theta_c)$. The controller knows a priori that the system is control-affine with relative degree two, but it is not aware of f and g . A zero-mean prior $\mathbf{M}_0(\mathbf{x}) = \mathbf{0}_{n \times (1+m)}$ and randomly generated covariance matrices \mathbf{A} and \mathbf{B} are used to initialize the MVGP model. We formulate a SOCP as in (28) with $r = 2$. We specify a task requiring the pendulum to track a reference control signal $\hat{\pi}_k(\mathbf{x})$ and specify the optimization objective as $\|\mathbf{R}(\mathbf{x}_k)(\mathbf{u}_k - \hat{\pi}_k(\mathbf{x}_k))\|_2^2$ with $\mathbf{R}(\mathbf{x}_k) \equiv \mathbf{I}$. The reference signal $\hat{\pi}_k(\mathbf{x})$ is an ϵ -greedy policy [82], used to provide sufficient excitation in the training data. Concretely, $\hat{\pi}_k(\mathbf{x})$ is sampled from an ϵ_k weighted combination of Dirac delta distribution δ and Uniform distribution \mathbb{U} , $\hat{\pi}(\mathbf{x}_k) \sim (1 - \epsilon_k)\delta_{\mathbf{u}=0} + \epsilon_k\mathbb{U}[-20, 20]$, where $\epsilon_k = 10^{-k/50}$ so that ϵ_k goes from 1 to 0.01 in 100 steps. We initialize the system with parameters $\theta_0 = 75^\circ$, $\omega_0 = -0.01$, $\tau = 0.01$, $m = 1$, $g = 10$, $l = 1$. Our simulation results show that the pendulum remains in the safe region (see Fig. 6). Negative control inputs get rejected by the SOCP safety constraint, while positive inputs allow the pendulum to bounce back from the unsafe region.

B. Ackermann Vehicle

Consider an Ackermann-drive vehicle model with state $\mathbf{x} = [x, y, \theta]^\top$, including position (x, y) and orientation θ . To make the Ackermann dynamics affine in the control input, we define

$z \triangleq v \tan(\phi)$, where v is the linear velocity and ϕ is the steering angle. The Ackermann-drive dynamics are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}_{f(\mathbf{x})} + \underbrace{\begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & \frac{1}{L} \end{bmatrix}}_{g(\mathbf{x})} \underbrace{\begin{bmatrix} v \\ z \end{bmatrix}}_{\mathbf{u}}, \quad (37)$$

where L is the distance between the front and back wheels.

1) *Estimating Ackermann-drive Dynamics*: Similar to Sec. VIII-A1, we compare the computational efficiency and inference accuracy of MVGP and CoGP with *diagonal* and *full* covariance matrices on training data generated from the Ackermann-drive model in (37). We further explicitly assume translation-invariant dynamics, i.e., $F(\mathbf{x}) = F([0, 0, \theta]^\top)$, $\forall \mathbf{x} \in \mathbb{R}^3$. This assumption allows us to transfer the learned dynamics to unvisited parts of the environment. The results of the simulation are shown in Fig. 5. We again find that MVGP inference is significantly faster than CoGP. In terms of accuracy, MVGP is comparable to CoGP and restricting the covariance matrices to diagonal matrices increases the median variance-weighted error.

2) *Safe Control of Learned Ackermann-drive Dynamics*:

To test our safe controller on the Ackermann-drive vehicle, we consider navigation to a goal state in the presence of two circular obstacles in the environment. The CBF for circular obstacle $i \in \{1, 2\}$ with center $\mathbf{o}_i \in \mathbb{R}^2$ and radius $r_i > 0$ is:

$$h_i(\mathbf{x}) = q_1(\|\mathbf{x}_{1:2} - \mathbf{o}_i\|_2^2 - r_i^2) + q_2 \cos(\theta - \phi_o), \quad (38)$$

where $\phi_o = \tan^{-1}\left(\frac{y - \mathbf{o}_{i,2}}{x - \mathbf{o}_{i,1}}\right)$. We assume that a planning algorithm provides a desired time-parameterized trajectory $\mathbf{x}^{(d)} = [x^{(d)}, y^{(d)}, \theta^{(d)}]$ and its time derivative $\dot{\mathbf{x}}^{(d)}$. We select a control Lyapunov function candidate:

$$V(\mathbf{x}, \mathbf{x}^{(d)}) = \frac{w_1}{2}\rho^2 + w_2(1 - \cos(\alpha)), \quad (39)$$

where $\rho \triangleq \|\mathbf{x}_{1:2}^{(d)} - \mathbf{x}_{1:2}\|_2$ and $\alpha \triangleq \theta - \tan^{-1}\left(\frac{y^{(d)} - y}{x^{(d)} - x}\right)$. The control Lyapunov condition is given by:

$$\text{CLC}(\mathbf{x}, \mathbf{u}) = \nabla_{\mathbf{x}}^\top V(\mathbf{x}, \mathbf{x}^{(d)})F(\mathbf{x})\mathbf{u} + \nabla_{\mathbf{x}^{(d)}}^\top V(\mathbf{x}, \mathbf{x}^{(d)})\dot{\mathbf{x}}^{(d)} + \gamma_V V(\mathbf{x}, \mathbf{x}^{(d)}). \quad (40)$$

The control Barrier condition for each obstacle i is:

$$\text{CBC}(\mathbf{x}, \mathbf{u}; i) = \nabla_{\mathbf{x}}^\top h_i(\mathbf{x})F(\mathbf{x})\mathbf{u} + \gamma_{h_i} h_i(\mathbf{x}). \quad (41)$$

We implement the SOCP controller in Prop. 4 with parameters: $q_1 = 0.7$, $q_2 = 0.3$, $w_1 = 0.9$, $w_2 = 1.5$, $\gamma_{h_i} = 5$, $\gamma_V = 10$, $\tau = 0.01$.

We perform two experiments to evaluate the advantages of our model. First, we evaluate the importance of *accounting for the variance* of the dynamics estimate for safe control. Second, we evaluate the importance of *online learning* in reducing the variance and ensuring safety.

Recall the observation in Prop. 4 that $c(\tilde{p}_k = 0.5) = 0$. In this case, the SOCP in (19) reduces to the deterministic-case QP in (4). We dub the case of $\tilde{p}_k = 0.5$ as *Mean CBF* and the case of $\tilde{p}_k = 0.99$ as *Bayes CBF*. In both the cases, the covariance matrices are initialized as $\mathbf{A} = 0.01\mathbf{I}_3$ and $\mathbf{B} = \mathbf{I}_3$. The true dynamics, $F(\mathbf{x})$, is specified with $L = 12$, while the mean dynamics $\mathbf{M}_0(\mathbf{x})$ is obtained with $L = 1$. The result

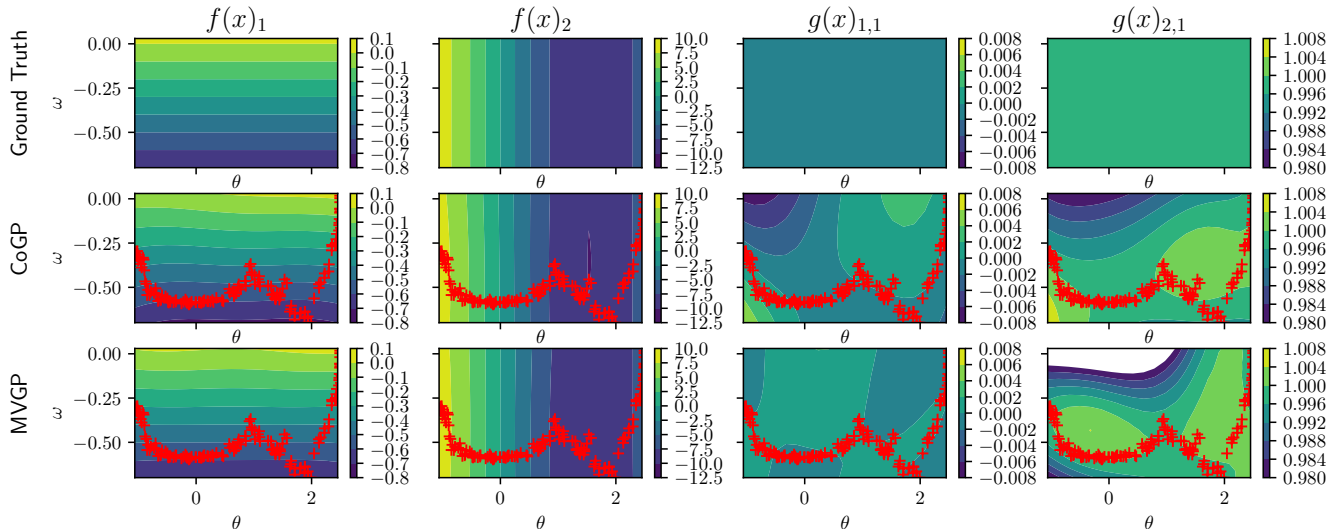


Fig. 4. Qualitative comparison of our Matrix Variate Gaussian Process (MVGP) regression with the Coregionalization GP (CoGP) model [65]. Training samples are generated using random control inputs to the pendulum. The training samples are shown as $+$. The learned models are evaluated on a 20×20 grid, shown as contour plots. The MVGP model has lower computational complexity than CoGP ($O(k^3)$ vs $O(k^3 n^3)$, see Sec. IV) without significant drop in accuracy.

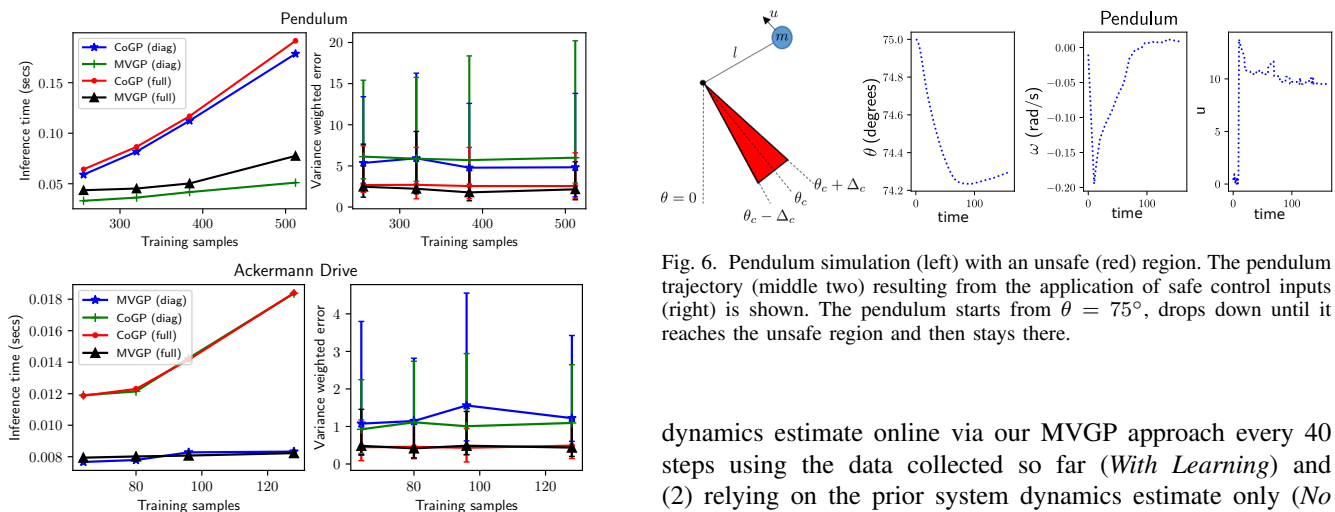


Fig. 5. Comparison of the computational efficiency and inference accuracy of MVGP and CoGP [65]. Both models are evaluated on the Pendulum and Ackermann systems in two modes: 1) with correlation matrix constrained to be diagonal (labelled *diag*) and 2) without any constraint on correlation matrix (labelled *full*). The results show that MVGP is significantly faster than CoGP. We argue in Sec. IV that the computational complexity of MVGP is $O(k^3)$ for k training examples, while that of CoGP is $O(k^3 n^3)$, where n is the state dimension. The *variance-weighted error* is computed via (36). The error bars denote the range from the 2nd to the 9th decile over 30 repetitions, centered around the median. The median errors of the MVGP and CoGP models are similar but enforcing diagonal covariance matrices, increases the median error.

of simulation is shown in Fig. 7. The *Bayes CBF* controller slows down when the safety constraint in (19d) hits zero and then changes direction while avoiding the obstacle. However, the *Mean CBF* controller is not able to avoid the obstacle because it does not take the variance of the dynamics estimate into account and the mean dynamics are inaccurate.

We also evaluate the importance of *online learning* for safe control. We compare two setups: (1) updating the system

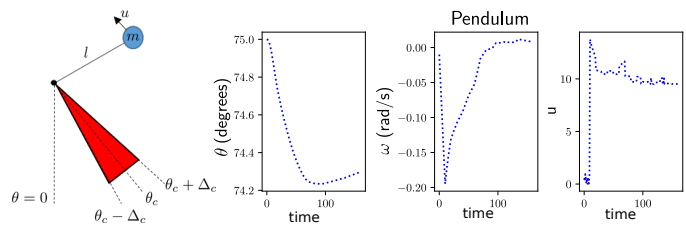


Fig. 6. Pendulum simulation (left) with an unsafe (red) region. The pendulum trajectory (middle two) resulting from the application of safe control inputs (right) is shown. The pendulum starts from $\theta = 75^\circ$, drops down until it reaches the unsafe region and then stays there.

dynamics estimate online via our MVGP approach every 40 steps using the data collected so far (*With Learning*) and (2) relying on the prior system dynamics estimate only (*No Learning*). In both the cases, we choose $\tilde{p}_k = 0.99$. The true dynamics are specified with $L = 1$, while prior mean dynamics correspond to $L = 8$. The covariance matrices \mathbf{A} and \mathbf{B} are initialized with elements independently sampled from a standard Gaussian distribution while ensuring them to be positive semi-definite as before. The result are shown in Fig. 8. Online learning of the vehicle dynamics reduces the variance and allows the vehicle to pass between the two obstacles. This is not possible using the prior dynamics distribution because the large variance makes the safety constraint in (19d) conservative.

3) *Evaluating triggering time*: We evaluate the bounds on the Lipschitz constant L_{f_k} (Lemma 1) and triggering time τ_k (Prop 5) for the Ackermann vehicle. At any time k , we use the following choice of parameters to compute τ_k , $\delta_L = 10^{-4}$, $\zeta = 10^{-2}$, $L_\alpha = 1$, and $L_{h_k} = \max_{\mathbf{x} \in \mathcal{X}_k} \nabla_{\mathbf{x}} h(\mathbf{x})$. We define \mathcal{X}_k to be a cuboid with sides $(0.2, 0.2, \frac{\pi}{50})$ in the directions x, y and θ respectively centered around the current state \mathbf{x}_k . We compute the Lipschitz constant L_{f_k} and τ_k both numerically and analytically. The numerical computation of L_{i,∂_j} is done

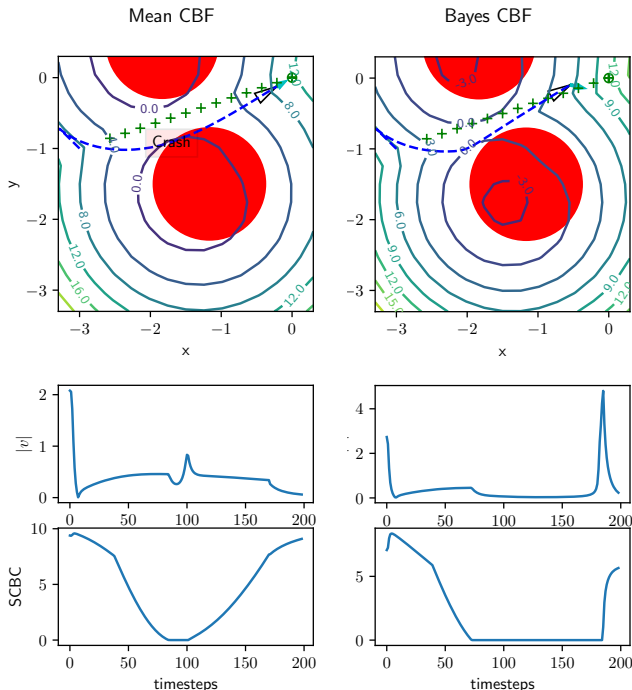


Fig. 7. Comparison of enforcing CBF constraints with Ackermann dynamics when accounting (*Bayes CBF*) and not accounting (*Mean CBF*) for the variance in the dynamics estimate. The top row shows the Ackermann vehicle trajectory in dashed blue with two obstacles in red. The contour plots shows the minimum of the SCBC (19d) values corresponding to the two obstacles, evaluated on the (x, y) grid while keeping θ and \mathbf{u} fixed. The middle row shows the magnitude of the velocity input over time. The bottom row shows the minimum of the two SCBC (19d) values over time. Enforcing safety using only the mean CBC (*Mean CBF*) results in a collision, while accounting for stochastic CBC (*Bayes CBF*) constraint causes the Ackermann vehicle to slow down and turn away from the unsafe region. A video rendering of these simulations is available in the supplementary material.

by sampling the GP, $\frac{\partial \mathbf{f}_i(\mathbf{x})}{\partial \mathbf{x}_j}$ over a $10 \times 10 \times 10$ grid in \mathcal{X}_k and taking the maximum norm as the Lipschitz constant. The rest of the computation for the numerical approximation of τ_k is the same as the analytical τ_k . The results are shown in Fig. 2. The shapes of the numerical and analytical Lipschitz constants are the same, although they differ by a couple orders of magnitude in scale since the analytical bound is conservative. Also, note that we are that in the numerical approximation δ_L is determined by the samples drawn from the GP. We also observe that the second term in the expression for L_{i,∂_j} (Lemma 1) dominates the effect of δ_L on the Lipschitz constant.

IX. CONCLUSION

Allowing artificial systems to safely adapt their own models during online operation will have significant implications for their successful use in unstructured changing real-world environments. This paper developed a Bayesian inference approach to approximate system dynamics and their uncertainty from online observations. The posterior distribution over the dynamics was used to specify probabilistic constraints that guarantee safe and stable online operation with high probability. Our results offer a promising approach for controlling complex

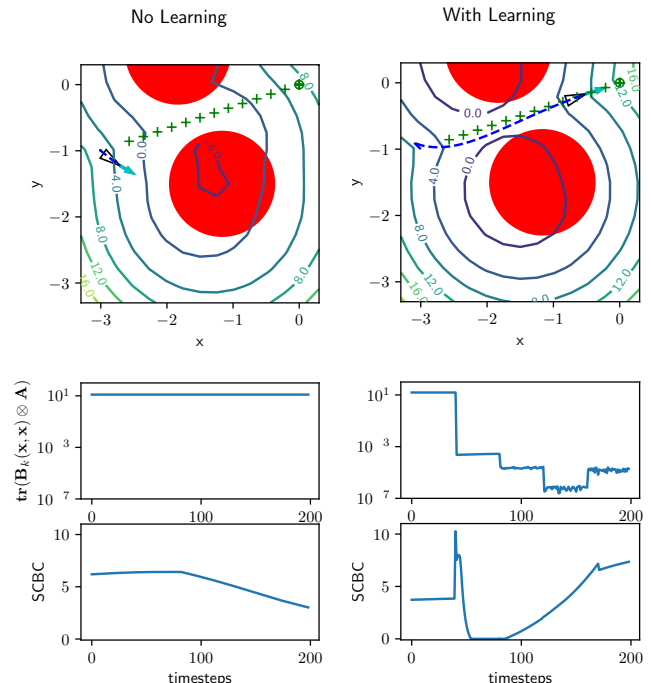


Fig. 8. The effect of online dynamics learning (right) versus no online learning (left) on the safe control of an Ackermann vehicle. The top row shows the vehicle trajectory in dashed blue with two obstacles in red. The middle row shows the trace of the covariance matrix $\text{tr}(\mathbf{B}_k(\mathbf{x}, \mathbf{x}) \otimes \mathbf{A})$, which we use as a measure of uncertainty. The bottom row shows the minimum of the two probabilistic safety constraint over time, as defined in (19d). Note that without learning, the vehicle gets stuck between the two obstacles because the uncertainty in the dynamics is too high, i.e., the safety condition in (19d) cannot be rendered positive. With online learning, however, the uncertainty is reduced enough to allow the safety condition to become positive in the area between the two obstacles. The dynamics distribution is updated every 40 time steps. Note the drop in uncertainty in the middle row at these time steps. A video rendering of these simulations is available in the supplementary material.

systems in challenging environments. Future work will focus on applications of the proposed approach to real robot systems.

APPENDIX A

ADDITIONAL PROPERTIES OF THE MVG DISTRIBUTION

Lemma 7 ([66]). *Let \mathbf{X} follow an MVG distribution $\mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$. Then, $\text{vec}(\mathbf{X}) \sim \mathcal{N}(\text{vec}(\mathbf{M}), \mathbf{B} \otimes \mathbf{A})$.*

Lemma 8 ([66]). *Let \mathbf{X} follow an MVG distribution $\mathcal{MN}(\mathbf{M}, \mathbf{A}, \mathbf{B})$ and let $\mathbf{C} \in \mathbb{R}^{d \times n}$ and $\mathbf{D} \in \mathbb{R}^{m \times d}$. Then,*

$$\begin{aligned} \mathbf{CX} &\sim \mathcal{MN}(\mathbf{CM}, \mathbf{CAC}^\top, \mathbf{B}), \\ \mathbf{XD} &\sim \mathcal{MN}(\mathbf{MD}, \mathbf{A}, \mathbf{D}^\top \mathbf{BD}). \end{aligned} \quad (42)$$

APPENDIX B

PROOF OF PROPOSITION 3

Let $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{K}_k(\mathbf{x}, \mathbf{x}'))$ be the posterior distribution of $\text{vec}(F(\mathbf{x}))$ conditioned on the training data $(\mathbf{X}_{1:k}, \mathbf{U}_{1:k}, \mathbf{X}_{1:k})$. The mean and variance can be obtained by applying Schur's complement to (8),

$$\begin{aligned} \text{vec}(\mathbf{M}_k(\mathbf{x})) &= \text{vec}(\mathbf{M}_0(\mathbf{x})) + \\ &(\underline{\mathbf{B}}_{1:k}(\mathbf{x}) \otimes \mathbf{A})^\top (\underline{\mathbf{B}}_{1:k}^{1:k} \otimes \mathbf{A})^{-1} \text{vec}(\dot{\mathbf{X}} - \mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}), \end{aligned} \quad (43)$$

$$\mathbf{K}_k(\mathbf{x}, \mathbf{x}) = \mathbf{B}_0(\mathbf{x}, \mathbf{x}) \otimes \mathbf{A} - (\underline{\mathbf{B}}_{1:k}(\mathbf{x}) \otimes \mathbf{A})^\top (\underline{\mathbf{B}}_{1:k}^{1:k} \otimes \mathbf{A})^{-1} (\underline{\mathbf{B}}_{1:k}^\top(\mathbf{x}) \otimes \mathbf{A}). \quad (44)$$

where $\underline{\mathbf{B}}_{1:k}^{1:k} \triangleq \underline{\mathbf{U}}_{1:k}^\top \underline{\mathbf{B}}_{1:k}^{1:k} \underline{\mathbf{U}}_{1:k} + \sigma^2 \mathbf{I}_k$ and $\underline{\mathbf{B}}_{1:k}(\mathbf{x}) \triangleq \underline{\mathbf{B}}_{1:k}(\mathbf{x}) \underline{\mathbf{U}}_{1:k}$. For appropriately sized matrices \mathbf{P} , \mathbf{Q} , \mathbf{R} , \mathbf{S} , the Kronecker product satisfies $(\mathbf{P} \otimes \mathbf{Q})(\mathbf{R} \otimes \mathbf{S}) = (\mathbf{P}\mathbf{R} \otimes \mathbf{Q}\mathbf{S})$ and $(\mathbf{P} \otimes \mathbf{Q})^{-1} = \mathbf{P}^{-1} \otimes \mathbf{Q}^{-1}$. Thus, we can rewrite the mean as:

$$\begin{aligned} \text{vec}(\mathbf{M}_k(\mathbf{x})) &= \text{vec}(\mathbf{M}_0(\mathbf{x})) + \\ &\left((\underline{\mathbf{B}}_{1:k}^\top(\mathbf{x}) [\underline{\mathbf{B}}_{1:k}^{1:k}]^{-1}) \otimes \mathbf{A} \mathbf{A}^{-1} \right) \text{vec}(\dot{\mathbf{X}}_{1:k} - \mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}). \end{aligned} \quad (45)$$

Applying $(\mathbf{P} \otimes \mathbf{Q}) \text{vec}(\mathbf{R}) = \text{vec}(\mathbf{Q}\mathbf{R}\mathbf{P}^\top)$, we get

$$\begin{aligned} \mathbf{M}_k(\mathbf{x}) &= \mathbf{M}_0(\mathbf{x}) + (\dot{\mathbf{X}}_{1:k} - \mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}) [\underline{\mathbf{B}}_{1:k}^{1:k}]^{-1} \underline{\mathbf{B}}_{1:k}^\top(\mathbf{x}) \\ &= \mathbf{M}_0(\mathbf{x}) + (\dot{\mathbf{X}}_{1:k} - \mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}) (\underline{\mathbf{U}}_{1:k} \underline{\mathbf{B}}_{1:k}(\mathbf{x}))^\dagger. \end{aligned} \quad (46)$$

Similarly, the covariance can be rewritten as,

$$\begin{aligned} \mathbf{K}_k(\mathbf{x}, \mathbf{x}') &= \left(\mathbf{B}_0(\mathbf{x}, \mathbf{x}') - \underline{\mathbf{B}}_{1:k}(\mathbf{x}) [\underline{\mathbf{B}}_{1:k}^{1:k}]^{-1} \underline{\mathbf{B}}_{1:k}^\top(\mathbf{x}') \right) \otimes \mathbf{A} \\ &= \left(\mathbf{B}_0(\mathbf{x}, \mathbf{x}') - \underline{\mathbf{B}}_{1:k}(\mathbf{x}') \underline{\mathbf{U}}_{1:k} (\underline{\mathbf{U}}_{1:k} \underline{\mathbf{B}}_{1:k}(\mathbf{x}))^\dagger \right) \otimes \mathbf{A}. \end{aligned} \quad (47)$$

Defining $\mathbf{B}_k(\mathbf{x}, \mathbf{x}')$ such that $\mathbf{K}_k(\mathbf{x}, \mathbf{x}) = \mathbf{B}_k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A}$, we can write the posterior distribution of $\text{vec}(F(\mathbf{x}))$ as $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{B}_k(\mathbf{x}, \mathbf{x}') \otimes \mathbf{A})$. \square

APPENDIX C

COREGIONALIZATION GAUSSIAN PROCESS

Here, we show how the Coregionalization model [65] can be applied to Gaussian process inference when the training data is available as a matrix vector product.

Lemma 9. Let $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\mathbf{M}_0(\mathbf{x})), \Sigma \kappa_0(\mathbf{x}, \mathbf{x}'))$, where $\Sigma \in \mathbb{R}^{(1+m)n \times (1+m)n}$ is the covariance matrix of the output dimensions, and $\kappa_0(\mathbf{x}, \mathbf{x}')$ is the kernel function. Denote the kernel function evaluation over the training data as,

$$\begin{aligned} \underline{\mathbf{K}}_{1:k}^{1:k} &\triangleq \begin{bmatrix} \kappa_0(\mathbf{x}_1, \mathbf{x}_1) & \dots & \kappa_0(\mathbf{x}_k, \mathbf{x}_1) \\ \vdots & & \vdots \\ \kappa_0(\mathbf{x}_k, \mathbf{x}_1) & \dots & \kappa_0(\mathbf{x}_k, \mathbf{x}_k) \end{bmatrix} \in \mathbb{R}^{k \times k} \\ \underline{\mathbf{K}}_{1:k}(\mathbf{x}) &\triangleq [\kappa_0(\mathbf{x}_1, \mathbf{x}) \quad \dots \quad \kappa_0(\mathbf{x}_k, \mathbf{x})] \in \mathbb{R}^{1 \times k}. \end{aligned} \quad (48)$$

Let $\mathcal{M}_{1:k}$ and $\underline{\mathbf{U}}_{1:k}$ be defined as in Proposition 3. Then, the posterior distribution of $\text{vec}(F(\mathbf{x}))$, given the training data $(\dot{\mathbf{X}}_{1:k}, \mathbf{X}_{1:k}, \underline{\mathbf{U}}_{1:k})$, is $\mathcal{GP}(\text{vec}(\mathbf{M}_k(\mathbf{x})), \mathbf{K}_k(\mathbf{x}, \mathbf{x}'))$, where

$$\begin{aligned} \text{vec}(\mathbf{M}_k(\mathbf{x})) &\triangleq \text{vec}(\mathbf{M}_0(\mathbf{x})) + \\ &\underline{\mathbf{K}}_{1:k}(\mathbf{x}) [\underline{\mathbf{K}}_{1:k}^{1:k}]^{-1} \text{vec}(\dot{\mathbf{X}}_{1:k} - \mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}), \\ \mathbf{K}_k(\mathbf{x}, \mathbf{x}') &\triangleq \Sigma \kappa_0(\mathbf{x}, \mathbf{x}') - \underline{\mathbf{K}}_{1:k}(\mathbf{x}) [\underline{\mathbf{K}}_{1:k}^{1:k}]^{-1} \underline{\mathbf{K}}_{1:k}^\top(\mathbf{x}'), \end{aligned} \quad (49)$$

$\underline{\mathbf{K}}_{1:k}^{1:k} \triangleq (\underline{\mathbf{U}}_{1:k}^\top \otimes \mathbf{I}_n) (\underline{\mathbf{K}}_{1:k}^{1:k} \otimes \Sigma) (\underline{\mathbf{U}}_{1:k} \otimes \mathbf{I}_n) + \mathbf{I}_k \otimes \mathbf{S}$, and $\underline{\mathbf{K}}_{1:k}(\mathbf{x}) \triangleq (\underline{\mathbf{K}}_{1:k}(\mathbf{x}) \otimes \Sigma) (\underline{\mathbf{U}}_{1:k} \otimes \mathbf{I}_n)$.

Proof. Using $\text{vec}(\mathbf{P}\mathbf{Q}\mathbf{R}) = (\mathbf{R}^\top \otimes \mathbf{P}) \text{vec}(\mathbf{Q})$, we can rewrite $F(\mathbf{x})\underline{\mathbf{u}}$ as

$$F(\mathbf{x})\underline{\mathbf{u}} = \text{vec}(\mathbf{I}_n F(\mathbf{x})\underline{\mathbf{u}}) = (\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n) \text{vec}(F(\mathbf{x})). \quad (50)$$

Thus, the variance of $F(\mathbf{x})\underline{\mathbf{u}}$ can be expressed in terms of the kernel $\Sigma \kappa_0(\mathbf{x}, \mathbf{x}')$:

$$F(\mathbf{x})\underline{\mathbf{u}} \sim \mathcal{N}(\mathbf{M}_0(\mathbf{x})\underline{\mathbf{u}}, (\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n) \Sigma \kappa_0(\mathbf{x}, \mathbf{x}') (\underline{\mathbf{u}} \otimes \mathbf{I}_n)). \quad (51)$$

The training data is generated from $\dot{\mathbf{x}} = F(\mathbf{x})\underline{\mathbf{u}} + \mathbf{w}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{S})$ and is jointly Gaussian:

$$\text{vec}(\dot{\mathbf{X}}_{1:k}) \sim \mathcal{N}(\text{vec}(\mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}), \underline{\mathbf{K}}_{1:k}^{1:k}), \quad (52)$$

The covariance for a single point is given by,

$$\begin{aligned} \text{cov}(\text{vec}(F(\mathbf{x})), F(\mathbf{x}')\underline{\mathbf{u}}) \\ = \text{cov}(\text{vec}(F(\mathbf{x})), (\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n) \text{vec}(F(\mathbf{x}')))) = \Sigma \kappa_0(\mathbf{x}, \mathbf{x}') (\underline{\mathbf{u}} \otimes \mathbf{I}_n). \end{aligned} \quad (53)$$

A similar expression for covariance can be obtained between the training data and the test data, $\text{cov}(\text{vec}(F(\mathbf{x})), \dot{\mathbf{X}}_{1:k}) = \underline{\mathbf{K}}_{1:k}(\mathbf{x})$. We can now write the joint distribution between the training and test data,

$$\begin{bmatrix} \text{vec}(\dot{\mathbf{X}}_{1:k}) \\ \text{vec}(F(\mathbf{x})) \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \text{vec}(\mathcal{M}_{1:k} \underline{\mathbf{U}}_{1:k}) \\ \mathbf{M}_0(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \underline{\mathbf{K}}_{1:k}^{1:k} & \underline{\mathbf{K}}_{1:k}^\top(\mathbf{x}') \\ \underline{\mathbf{K}}_{1:k}(\mathbf{x}) & \Sigma \kappa_0(\mathbf{x}, \mathbf{x}') \end{bmatrix} \right).$$

Applying a Schur complement, we get the posterior distribution in (49). \square

APPENDIX D PROOF OF LEMMA 1

Proof. The bound on each element of the Jacobian of $\mathbf{f}(\mathbf{x})$ is due to [69, Lemma B.2] with probability at least $1 - \frac{\delta_L}{n^2}$,

$$|\mathbf{f}_i(\mathbf{x}') - \mathbf{f}_i(\mathbf{x})| \leq L_{i,\partial_j} |\mathbf{x}'_j - \mathbf{x}_j| \quad \forall i, j \in \{1, \dots, n\}. \quad (54)$$

Denote the event in (54) by $\mathcal{E}_{i,\partial_j}$ so that $\mathbb{P}(\mathcal{E}_{i,\partial_j}) \geq 1 - \frac{\delta_L}{n^2}$. A lower bound on the probability of intersection of all events can be computed using a union bound:

$$\begin{aligned} \mathbb{P}(\cap_{i=1}^n \cap_{j=1}^n \mathcal{E}_{i,\partial_j}) &= 1 - \mathbb{P}(\cup_{i=1}^n \cup_{j=1}^n \mathcal{E}_{i,\partial_j}^c) \\ &\geq 1 - \sum_{i=1}^n \sum_{j=1}^n \mathbb{P}(\mathcal{E}_{i,\partial_j}^c) = 1 - \sum_{i=1}^n \sum_{j=1}^n (1 - \mathbb{P}(\mathcal{E}_{i,\partial_j})) \\ &\geq 1 - \delta_L. \end{aligned} \quad (55)$$

Adding (54) for all $j \in \{1, \dots, n\}$ and using the Cauchy-Schwarz inequality, we get for each $i \in \{1, \dots, n\}$:

$$n |\mathbf{f}_i(\mathbf{x}') - \mathbf{f}_i(\mathbf{x})| \leq \sum_{j=1}^n L_{i,\partial_j} |\mathbf{x}'_j - \mathbf{x}_j| \leq \sqrt{\sum_{j=1}^n L_{i,\partial_j}^2} \|\mathbf{x}' - \mathbf{x}\|_2.$$

Squaring and adding all inequalities for $i \in \{1, \dots, n\}$, we get:

$$\|\mathbf{f}(\mathbf{x}') - \mathbf{f}(\mathbf{x})\|_2 \leq \sqrt{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n L_{i,\partial_j}^2} \|\mathbf{x}' - \mathbf{x}\|_2. \quad \square$$

APPENDIX E PROOF OF LEMMA 4

Recall the definition of $\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})$:

$$\begin{aligned} \text{CBC}^{(r)}(\mathbf{x}, \mathbf{u}) &= \mathcal{L}_f^{(r)} h(\mathbf{x}) + \mathcal{L}_g \mathcal{L}_f^{(r-1)} h(\mathbf{x}) \mathbf{u} + \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ &= \mathcal{L}_f [\mathcal{L}_f^{(r-1)} h(\mathbf{x})] + \mathcal{L}_g [\mathcal{L}_f^{(r-1)} h(\mathbf{x})] \mathbf{u} + \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ &= \nabla_{\mathbf{x}} [\mathcal{L}_f^{(r-1)} h(\mathbf{x})]^\top f(\mathbf{x}) + \nabla_{\mathbf{x}} [\mathcal{L}_f^{(r-1)} h(\mathbf{x})]^\top g(\mathbf{x}) \mathbf{u} + \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ &= \nabla_{\mathbf{x}} [\mathcal{L}_f^{(r-1)} h(\mathbf{x})]^\top F(\mathbf{x}) \underline{\mathbf{u}} + \mathbf{k}_\alpha^\top \eta(\mathbf{x}). \end{aligned}$$

Since, $\underline{\mathbf{u}} = [1, \mathbf{u}^\top]^\top$, we can rewrite the above as

$$\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u}) = \left(\nabla_{\mathbf{x}} \mathcal{L}_f^{(r-1)} h(\mathbf{x})^\top F(\mathbf{x}) + \begin{bmatrix} \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ \mathbf{0}_m \end{bmatrix}^\top \right) \underline{\mathbf{u}}$$

Using linearity of expectation, we see that $\mathbb{E}[\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})]$ is linear in \mathbf{u} :

$$\mathbb{E}[\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})] = \underbrace{\left(\mathbb{E}[F(\mathbf{x})^\top \nabla_{\mathbf{x}} \mathcal{L}_f^{(r-1)} h(\mathbf{x})] + \mathbb{E} \left[\begin{bmatrix} \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ \mathbf{0}_m \end{bmatrix} \right] \right)^\top}_{\mathbf{e}^{(r)}(\mathbf{x})} \mathbf{u}. \quad (56)$$

Also, applying $\text{Var}(\mathbf{x}^\top \mathbf{u}) = \text{Var}(\mathbf{u}^\top \mathbf{x}) = \mathbf{u}^\top \text{Var}(\mathbf{x}) \mathbf{u}$ to $\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})$, we get:

$$\text{Var}[\text{CBC}^{(r)}(\mathbf{x}, \mathbf{u})] = \mathbf{u}^\top \underbrace{\text{Var} \left[\nabla_{\mathbf{x}} \mathcal{L}_f^{(r-1)} h(\mathbf{x})^\top F(\mathbf{x}) + \begin{bmatrix} \mathbf{k}_\alpha^\top \eta(\mathbf{x}) \\ \mathbf{0}_m \end{bmatrix} \right]^\top}_{\mathbf{V}^{(r)}(\mathbf{x}) \mathbf{V}^{(r)}(\mathbf{x})^\top} \mathbf{u}. \quad (57) \quad \square$$

APPENDIX F PROOF OF PROPOSITION 6

Unlike CBC for relative degree one, the distribution of $\text{CBC}^{(r)}$ is not a Gaussian Process for $r \geq 2$. Hence, instead of computing the probability distribution analytically, we use Cantelli's inequality to bound the mean and variance of $\text{CBC}_k^{(r)}$. For any scalar $\lambda > 0$, we have

$$\mathbb{P}(\text{CBC}_k^{(r)} \geq \mathbb{E}[\text{CBC}_k^{(r)}] - \lambda \mid \mathbf{x}_k, \mathbf{u}_k) \geq 1 - \frac{\text{Var}[\text{CBC}_k^{(r)}]}{\text{Var}[\text{CBC}_k^{(r)}] + \lambda^2}.$$

Since we want the probability to be greater than \tilde{p}_k , we ensure its lower bound is greater than \tilde{p}_k , i.e.,

$$1 - \frac{\text{Var}[\text{CBC}_k^{(r)}]}{\text{Var}[\text{CBC}_k^{(r)}] + \lambda^2} \geq \tilde{p}_k. \quad (58)$$

The terms can be rearranged into,

$$(1 - \tilde{p}_k)(\text{Var}[\text{CBC}_k^{(r)}] + \lambda^2) \geq \text{Var}[\text{CBC}_k^{(r)}] \\ \lambda \geq \sqrt{\frac{\tilde{p}_k}{1 - \tilde{p}_k} \text{Var}[\text{CBC}_k^{(r)}]}. \quad (59)$$

For some desired margin, $\mathbb{E}[\text{CBC}_k^{(r)}] \geq \zeta$, we can substitute $\lambda = (\mathbb{E}[\text{CBC}_k^{(r)}] - \zeta) > 0$ in (59), so that:

$$\mathbb{E}[\text{CBC}_k^{(r)}] - \zeta > \sqrt{\frac{\tilde{p}_k}{1 - \tilde{p}_k}} \text{Var}^{\frac{1}{2}}[\text{CBC}_k^{(r)}] \\ \implies \mathbb{P}(\text{CBC}_k^{(r)} \geq \zeta \mid \mathbf{x}_k, \mathbf{u}_k) \geq \tilde{p}_k \quad (60)$$

Using the constraint in (24) and the explicit expressions in (56) and (57) for the mean and variance of $\text{CBC}_k^{(r)}$, we get:

$$\pi(\mathbf{x}_k) \in \arg \min_{\mathbf{u}_k \in \mathcal{U}} \|\mathbf{R}(\mathbf{x}_k) \mathbf{u}_k\|_2 \quad (61) \\ \text{s.t. } \mathbf{e}_k^{(r)}(\mathbf{x}_k)^\top \mathbf{u}_k - \zeta - c^{(r)}(\tilde{p}_k) \left\| \mathbf{V}_k^{(r)}(\mathbf{x}_k) \mathbf{u}_k \right\|_2 \geq 0,$$

where $c^{(r)}(\tilde{p}_k) = \sqrt{\frac{\tilde{p}_k}{1 - \tilde{p}_k}}$. To ensure that (61) is a standard SOCP, we convert the objective into a SOCP constraint by introducing an auxiliary variable y , leading to (28). \square

APPENDIX G PROOF OF LEMMA 5

Recall the following result about the mean and cumulants of a quadratic form of Gaussian random vectors.

Lemma 10 ([83, p. 55]). *Let \mathbf{x} be a Gaussian random vector with mean $\bar{\mathbf{x}}$ and covariance matrix Σ . Let Λ be a symmetric matrix and consider the random variable $\mathbf{x}^\top \Lambda \mathbf{x}$. The mean of $\mathbf{x}^\top \Lambda \mathbf{x}$ is*

$$\mathbb{E}[\mathbf{x}^\top \Lambda \mathbf{x}] = \bar{\mathbf{x}}^\top \Lambda \bar{\mathbf{x}} + \text{tr}(\Lambda \Sigma).$$

The r th cumulant of $\mathbf{x}^\top \Lambda \mathbf{x}$ is:

$$\mathcal{K}_r(\mathbf{x}^\top \Lambda \mathbf{x}) = 2^{r-1} (r-1)! [\text{tr}(\Lambda \Sigma)^r + r \bar{\mathbf{x}}^\top \Lambda (\Sigma \Lambda)^{r-1} \bar{\mathbf{x}}].$$

The variance of $\mathbf{x}^\top \Lambda \mathbf{x}$ is the second cumulant:

$$\text{Var}[\mathbf{x}^\top \Lambda \mathbf{x}] = \mathcal{K}_2(\mathbf{x}^\top \Lambda \mathbf{x}) = 2 \text{tr}(\Lambda \Sigma)^2 + 4 \bar{\mathbf{x}}^\top \Lambda \Sigma \Lambda \bar{\mathbf{x}}.$$

The covariance between \mathbf{x} and $\mathbf{x}^\top \Lambda \mathbf{x}$ is:

$$\text{cov}(\mathbf{x}, \mathbf{x}^\top \Lambda \mathbf{x}) = 2 \Sigma \Lambda \bar{\mathbf{x}}$$

Returning to the proof of Lemma 5, consider the three Gaussian random vectors \mathbf{x} , \mathbf{y} , and \mathbf{z} . Note that $\mathbf{x}^\top \mathbf{y}$ can be written as a quadratic form:

$$\mathbf{x}^\top \mathbf{y} = \frac{1}{2} \begin{bmatrix} \mathbf{x}^\top & \mathbf{y}^\top & \mathbf{z}^\top \end{bmatrix} \begin{bmatrix} 0 & \mathbf{I} & 0 \\ \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}. \quad (62)$$

Applying Lemma 10 to (62), shows that:

$$\mathbb{E}[\mathbf{x}^\top \mathbf{y}] = \bar{\mathbf{x}}^\top \bar{\mathbf{y}} + \frac{1}{2} \text{tr}(\text{cov}(\mathbf{x}, \mathbf{y}) + \text{cov}(\mathbf{y}, \mathbf{x})) \\ \text{Var}[\mathbf{x}^\top \mathbf{y}] = \frac{1}{2} \text{tr}(\text{cov}(\mathbf{x}, \mathbf{y}) + \text{cov}(\mathbf{y}, \mathbf{x}))^2 + \bar{\mathbf{y}}^\top \text{Var}[\mathbf{x}] \bar{\mathbf{y}} \\ + \bar{\mathbf{x}}^\top \text{Var}[\mathbf{y}] \bar{\mathbf{x}} + \bar{\mathbf{y}}^\top \text{cov}(\mathbf{x}, \mathbf{y}) \bar{\mathbf{x}} + \bar{\mathbf{x}}^\top \text{cov}(\mathbf{y}, \mathbf{x}) \bar{\mathbf{y}} \\ \begin{bmatrix} \text{cov}(\mathbf{x}, \mathbf{x}^\top \mathbf{y}) \\ \text{cov}(\mathbf{y}, \mathbf{x}^\top \mathbf{y}) \\ \text{cov}(\mathbf{z}, \mathbf{x}^\top \mathbf{y}) \end{bmatrix} = \text{cov} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}, \mathbf{x}^\top \mathbf{y} \right) \\ = 2 \begin{bmatrix} \text{Var}[\mathbf{x}] & \text{cov}(\mathbf{x}, \mathbf{y}) & \text{cov}(\mathbf{x}, \mathbf{z}) \\ \text{cov}(\mathbf{y}, \mathbf{x}) & \text{Var}[\mathbf{y}] & \text{cov}(\mathbf{y}, \mathbf{z}) \\ \text{cov}(\mathbf{z}, \mathbf{x}) & \text{cov}(\mathbf{z}, \mathbf{y}) & \text{Var}[\mathbf{z}] \end{bmatrix} \frac{1}{2} \begin{bmatrix} \bar{\mathbf{y}} \\ \bar{\mathbf{x}} \\ \mathbf{0} \end{bmatrix}. \quad \square$$

REFERENCES

- [1] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic Safety Constraints for Learned High Relative Degree System Dynamics," in *Learn. Dyn. Cont. (LADC)*, 2020.
- [2] K.-D. Kim and P. R. Kumar, "Cyber-physical systems: A perspective at the centennial," *Proc. of IEEE*, vol. 100, pp. 1287–1308, 2012.
- [3] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Cont. Sys.*, vol. 23, no. 2, pp. 20–33, 2003.
- [4] J. A. Bagnell, D. Bradley, D. Silver, B. Sofman, and A. Stentz, "Learning for autonomous navigation," *IEEE Robot. & Auto. Mag.*, vol. 17, no. 2, pp. 74–84, 2010.
- [5] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety Augmented Value Estimation from Demonstrations (SAVED): Safe Deep Model-Based RL for Sparse Cost Robotic Tasks," *IEEE Robotics and Auto. Lett.*, vol. 5, no. 2, pp. 3612–3619, 2020.
- [6] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 465–472.
- [7] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the Sample Complexity of the Linear Quadratic Regulator," *Foundations of Computational Mathematics*, vol. 20, pp. 633–679, 2020.

- [8] T. Sarkar and A. Rakhlin, "Near optimal finite time identification of arbitrary linear dynamical systems," in *Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 5610–5618.
- [9] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: in the shallows of the DeePC," in *European Control Conf. (ECC)*. IEEE, 2019, pp. 307–312.
- [10] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *American Control Conf. (ACC)*. IEEE, 2018, pp. 1520–1527.
- [11] M. J. Khojasteh, A. Khina, M. Franceschetti, and T. Javidi, "Learning-based Attacks in Cyber-Physical Systems," *IEEE Trans. Cont. Network Sys.*, 2020.
- [12] A. Liu, G. Shi, S.-J. Chung, A. Anandkumar, and Y. Yue, "Robust regression for safe exploration in control," in *Learn. Dyn. Cont. (LADC)*, 2020, pp. 608–619.
- [13] D. D. Fan, A.-a. Agha-mohammadi, and E. A. Theodorou, "Deep Learning Tubes for Tube MPC," *Robotics: Science and Systems*, 2020.
- [14] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using Gaussian processes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 537–550, 2014.
- [15] A. Lakshmanan, A. Gahlawat, and N. Hovakimyan, "Safe feedback motion planning: A contraction theory and L1-adaptive control based approach," in *IEEE Conf. Decision and Cont. (CDC)*, 2020, pp. 1578–1583.
- [16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [17] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, "Agile Autonomous Driving using End-to-End Deep Imitation Learning," in *Robotics: science and systems*, 2018.
- [18] G. Kahn, P. Abbeel, and S. Levine, "BADGR: An autonomous self-supervised learning-based navigation system," *IEEE Robot. Autom. Lett.*, 2020.
- [19] M. Ono, B. Rothrock, K. Otsu, S. Higa, Y. Iwashita, A. Didier, T. Islam, C. Laporte, V. Sun, K. Stack *et al.*, "MAARS: Machine learning-based Analytics for Automated Rover Systems," in *Aerospace Conf.* IEEE, 2020, pp. 1–17.
- [20] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 462–467, 2007.
- [21] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Auto. Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [22] X. Xu, T. Waters, D. Pickem, P. Glotfelter, M. Egerstedt, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds," in *IEEE Conf. on Cont. Tech. and App.(CCTA)*, 2017, pp. 1769–1775.
- [23] 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.
- [24] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Trans. Auto. Control*, vol. 52, no. 8, pp. 1415–1428, 2007.
- [25] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *European Control Conf. (ECC)*, June 2019, pp. 3420–3431.
- [26] F. S. Barbosa, L. Lindemann, D. V. Dimarogonas, and J. Tumova, "Provably safe control of lagrangian systems in obstacle-scattered environments," in *IEEE Conf. Decision and Cont. (CDC)*, 2020, pp. 2056–2061.
- [27] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.
- [28] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 757–762, 2019.
- [29] K. Garg and D. Panagou, "Control-Lyapunov and Control-Barrier Functions based Quadratic Program for Spatio-temporal Specifications," in *IEEE Conf. Decision and Cont. (CDC)*, 2019, pp. 1422–1429.
- [30] G. Yang, C. Belta, and R. Tron, "Self-triggered control for safety critical systems using control barrier functions," in *American Control Conf. (ACC)*. IEEE, 2019, pp. 4454–4459.
- [31] J. Umlauf and S. Hirche, "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Trans. Auto. Control*, vol. 65, no. 10, pp. 4154–4169, 2020.
- [32] F. Solowjow and S. Trimpe, "Event-triggered learning," *Automatica*, vol. 117, p. 109009, 2020.
- [33] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. Decision and Cont. (CDC)*, 2012, pp. 3270–3285.
- [34] K. Hashimoto, Y. Yoshimura, and T. Ushio, "Learning self-triggered controllers with Gaussian processes," *IEEE Transactions on Cybernetics*, 2020, to appear.
- [35] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conf. (ACC)*. IEEE, 2015, pp. 4542–4548.
- [36] Q. Nguyen and K. Sreenath, "Optimal robust control for constrained nonlinear hybrid systems with application to bipedal locomotion," in *American Control Conf. (ACC)*. IEEE, 2016, pp. 4807–4813.
- [37] Q. Nguyen and K. Sreenath, "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," in *American Control Conf. (ACC)*. IEEE, 2016, pp. 322–328.
- [38] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *IEEE Conf. Decision and Cont. (CDC)*, 2019, pp. 474–479.
- [39] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *IEEE Conf. Decision and Cont. (CDC)*, 2018, pp. 6059–6066.
- [40] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *IEEE Int Conf. Robot Autom. (ICRA)*, 2016, pp. 493–496.
- [41] 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. Auto. Control*, vol. 64, no. 7, pp. 2737–2752, 2018.
- [42] O. Bastani, "Safe reinforcement learning with nonlinear dynamics via model predictive shielding," in *American Control Conf. (ACC)*, 2021, pp. 3488–3494.
- [43] K. P. Wabersich and M. N. Zeilinger, "Safe exploration of nonlinear dynamical systems: A predictive safety filter for reinforcement learning," *arXiv preprint arXiv:1812.05506*, 2018.
- [44] E. Biyik, J. Margoliash, S. R. Alimo, and D. Sadigh, "Efficient and safe exploration in deterministic markov decision processes with unknown transition models," in *American Control Conf. (ACC)*. IEEE, 2019, pp. 1792–1799.
- [45] N. Jansen, B. Könighofer, S. Junges, and R. Bloem, "Shielded decision-making in MDPs," *arXiv preprint arXiv:1807.06096*, 2018.
- [46] T. Lew, R. Bonalli, and M. Pavone, "Chance-constrained sequential convex programming for robust trajectory optimization," in *European Control Conf. (ECC)*. IEEE, 2020, pp. 1871–1878.
- [47] A. Clark, "Control Barrier Functions for Complete and Incomplete Information Stochastic Systems," in *American Control Conf. (ACC)*. IEEE, 2019, pp. 2928–2935.
- [48] C. Santoyo, M. Dutreix, and S. Coogan, "A Barrier Function Approach to Finite-Time Stochastic System Verification and Control," *Automatica*, 2020.
- [49] M. Aloysius Pereira, Z. Wang, I. Exarchos, and E. A. Theodorou, "Safe Optimal Control Using Stochastic Barrier Functions and Deep Forward-Backward SDEs," *arXiv preprint arXiv:2009.01196*, 2020.
- [50] S. Yaghoubi, K. Majd, G. Fainekos, T. Yamaguchi, D. Prokhorov, and B. Hoxha, "Risk-bounded control using stochastic barrier functions," *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1831–1836, 2021.
- [51] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A. a. Agha-mohammadi, and E. A. Theodorou, "Bayesian Learning-Based Adaptive Control for Safety Critical Systems," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4093–4099.
- [52] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *IEEE Int Conf. Robot Autom. (ICRA)*, 2018, pp. 2460–2465.
- [53] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3387–3395.
- [54] Z. Marvi and B. Kiumarsi, "Safe Off-policy Reinforcement Learning Using Barrier Functions," in *American Control Conf. (ACC)*. IEEE, 2020, pp. 2176–2181.
- [55] A. J. Taylor and A. D. Ames, "Adaptive Safety with Control Barrier Functions," in *American Control Conf. (ACC)*, 2020, pp. 1399–1405.
- [56] 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.
- [57] I. Salehi, G. Yao, and A. P. Dani, "Active Sampling based Safe Identification of Dynamical Systems using Extreme Learning Machines and Barrier Certificates," in *IEEE Int Conf. Robot Autom. (ICRA)*, 2019, pp. 22–28.

- [58] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learn. Dyn. Cont. (LADC)*, 2020, pp. 708–717.
- [59] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.
- [60] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1050–1059.
- [61] J. Harrison, A. Sharma, and M. Pavone, "Meta-learning Priors for Efficient Online Bayesian Regression," in *Work. Alg. Found. Robotics*. Springer, 2018, pp. 318–337.
- [62] B. Efron and T. Hastie, *Computer age statistical inference*. Cambridge University Press, 2016, vol. 5.
- [63] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 908–918.
- [64] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *IEEE Conf. Decision and Cont. (CDC)*, 2020, pp. 3699–3704.
- [65] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, pp. 195–266, 2012.
- [66] S. Sun, C. Chen, and L. Carin, "Learning Structured Weight Uncertainty in Bayesian Neural Networks," in *Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2017, pp. 1283–1292.
- [67] C. Louizos and M. Welling, "Structured and efficient variational deep learning with matrix Gaussian posteriors," in *Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1708–1716.
- [68] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.
- [69] A. Lederer, J. Umlauf, and S. Hirche, "Uniform error bounds for Gaussian process regression with application to safe control," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 659–669.
- [70] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193–228, 1998.
- [71] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," Online: <http://www.gurobi.com>, 2020.
- [72] L. Vandenberghe, "The CVXOPT linear and quadratic cone program solvers," Online: <https://cvxopt.org/userguide/coneprog>, 2010.
- [73] 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 Conf. Decision and Cont. (CDC)*, 2021.
- [74] P. Akella, M. Ahmadi, R. M. Murray, and A. D. Ames, "Formal test synthesis for safety-critical autonomous systems based on control barrier functions," in *IEEE Conf. Decision and Cont. (CDC)*, 2020, pp. 790–795.
- [75] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning Control Barrier Functions from Expert Demonstrations," *arXiv preprint arXiv:2004.03315*, 2020.
- [76] M. Srinivasan, A. Dabholkar, S. Coogan, and P. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," *International Conference on Intelligent Robots and Systems*, 2020.
- [77] M. Sarkar, D. Ghose, and E. A. Theodorou, "High-Relative Degree Stochastic Control Lyapunov and Barrier Functions," *arXiv preprint arXiv:2004.03856*, 2020.
- [78] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, "Training neural network controllers using control barrier functions in the presence of disturbances," in *IEEE Int. Conf. Intel. Trans. Sys.*, 2020, pp. 1–6.
- [79] R. J. Adler, *The geometry of random fields*. Chichester : John Wiley and Sons, 1981.
- [80] J. Taylor, "STAT352, Spatial Statistics: Notes on Random Fields," Stanford University, http://statweb.stanford.edu/~jtaylor/courses/stats352/notes/random_fields.pdf, 2009.
- [81] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPpyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration," in *Adv. Neural Inf. Process. Syst.*, 2018.
- [82] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [83] S. R. Searle and M. H. Gruber, *Linear models*. John Wiley & Sons, 1971.



Vikas Dhiman is an Assistant Professor in the ECE department at the University of Maine. His works lie in the localization, mapping and control algorithms for applications in robotics. He was a Postdoctoral Researcher at the University of California, San Diego (2019-21). He graduated in Elec. Engg (2008) from Indian Institute of Technology, Roorkee, earned his MS (2014) from University at Buffalo, and received his PhD (2019) from the University of Michigan, Ann Arbor.



Mohammad Javad Khojasteh (S'14–M'21) did his undergraduate studies at Sharif University of Technology from which he received B.Sc. degrees in both Electrical Engineering and Mathematics, in 2015. He received the M.Sc. and Ph.D. degrees in Electrical and Computer Engineering from University of California San Diego (UCSD), La Jolla, CA, in 2017, and 2019, respectively. In 2020, he was a postdoctoral scholar with Center for Autonomous Systems and Technology (CAST) at California Institute of Technology (Caltech), and a visitor at NASA's Jet Propulsion Laboratory (JPL), where he worked with Team CoSTAR. Currently, he is a Postdoctoral Associate with Laboratory for Information and Decision Systems (LIDS) at Massachusetts Institute of Technology (MIT).



Massimo Franceschetti (M'98–SM'11–F'18) received the Laurea degree (with highest honors) in computer engineering from the University of Naples, Naples, Italy, in 1997, the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, in 1999, and 2003, respectively. He is Professor of Electrical and Computer Engineering at the University of California at San Diego (UCSD). Before joining UCSD, he was a postdoctoral scholar at the University of California at Berkeley for two years. He has held visiting positions at the Vrije Universiteit Amsterdam, the École Polytechnique Fédérale de Lausanne, and the University of Trento. His research interests are in physical and information-based foundations of communication and control systems. He was awarded the C. H. Wilts Prize in 2003 for best doctoral thesis in electrical engineering at Caltech; the S.A. Schelkunoff Award in 2005 for best paper in the IEEE Transactions on Antennas and Propagation, a National Science Foundation (NSF) CAREER award in 2006, an Office of Naval Research (ONR) Young Investigator Award in 2007, the IEEE Communications Society Best Tutorial Paper Award in 2010, and the IEEE Control theory society Ruberti young researcher award in 2012. He has been elected fellow of the IEEE in 2018 and became a Guggenheim fellow for the natural sciences: engineering in 2019.



Nikolay Atanasov (S'07–M'16) is an Assistant Professor of Electrical and Computer Engineering at the University of California San Diego. He obtained a B.S. degree in Electrical Engineering from Trinity College, Hartford, CT, in 2008 and M.S. and Ph.D. degrees in Electrical and Systems Engineering from the University of Pennsylvania, Philadelphia, PA, in 2012 and 2015, respectively. His research focuses on robotics, control theory, and machine learning, applied to active sensing using ground and aerial robots. He works on probabilistic environment models that unify geometry and semantics and on optimal control and reinforcement approaches for minimizing uncertainty in these models. Dr. Atanasov's work has been recognized by the Joseph and Rosaline Wolf award for the best Ph.D. dissertation in Electrical and Systems Engineering at the University of Pennsylvania in 2015 and the best conference paper award at the International Conference on Robotics and Automation in 2017.