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 $C$ by evaluating if the system trajectory remains away from the boundary of $C$. 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
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:
\[ \dot{x} = f(x) + g(x)u = [f(x) \quad g(x)] \begin{bmatrix} 1 \\ u \end{bmatrix} \triangleq F(x)u, \]
(1)
where \( x(t) \in \mathcal{X} \subset \mathbb{R}^n \) and \( u(t) \in \mathcal{U} \subset \mathbb{R}^m \) 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 \( x, u, \dot{x} \).

A. Notation

We use bold lower-case letters for vectors (\( x \)), bold capital letters for matrices (\( X \)), and calligraphic capital letters for sets (\( \mathcal{X} \)). The boundary of a set \( \mathcal{X} \) is denoted by \( \partial \mathcal{X} \). Let \( \text{vec}(X) \in \mathbb{R}^{nm} \) be the vectorization of \( X \in \mathbb{R}^{n \times m} \), obtained by stacking the columns of \( X \). 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:
\[ H_{x,y}h(x,y) \triangleq \begin{bmatrix} \frac{\partial^2 h}{\partial x_1 \partial y_1} & \ldots & \frac{\partial^2 h}{\partial x_1 \partial y_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 h}{\partial x_n \partial y_1} & \ldots & \frac{\partial^2 h}{\partial x_n \partial y_m} \end{bmatrix} \]
\[ J_{x}f(x) \triangleq \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}. \]
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 \( C^r(\mathcal{X}, \mathbb{R}) \).

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 \( V \in C^1(\mathcal{X}, \mathbb{R}) \) is a control Lyapunov function (CLF) for the system in (1) if it is positive definite, \( V(x) > 0 \), \( \forall x \in \mathcal{X} \setminus \{0\} \), \( V(0) = 0 \), and satisfies:
\[ \inf_{u \in \mathcal{U}} \text{CLC}(x, u) \leq 0, \quad \forall x \in \mathcal{X}, \]
where \( \text{CLC}(x, u) \triangleq \mathcal{L}_f V(x) + \mathcal{L}_g V(x)u + \gamma(V(x)) \) is a control Lyapunov condition (CLC) defined for some class \( K \) function \( \gamma \).
Proposition 1 (Sufficient Condition for Stability [25]). If there exists a CLF $V(x)$ for system (1), then any Lipschitz continuous control policy $\pi(x) \in \{u \in U \mid CLC(x, u) \leq 0\}$ asymptotically stabilizes the system.

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

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

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

(3)

where $CBC(x, u) \triangleq L_I h(x) + L_g h(x) + \alpha(h(x))$ is a control barrier condition (CBC) defined for some extended class $K_\infty$ function $\alpha$.

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

Prop. 1 and Prop. 2 provide sufficient conditions for a control policy $\pi(x)$ applied to the system in (1) to guarantee stability and safety. Note that the conditions are defined by affine constraints in $u$. 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:

$$\pi(x) \in \arg \min_{u \in \mathcal{U}, \delta \in \mathbb{R}} \|R(x)u\|^2 + \lambda \delta^2$$

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

(4)

where $R(x) \in \mathbb{R}^{m \times m}$ is a matrix penalizing control effort and $\delta$ 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 $\hat{\pi}(x)$ is already available, it may be modified minimally online to ensure safety:

$$\pi(x) \in \arg \min_{u \in \mathcal{U}} \|R(x)(u - \hat{\pi}(x))\|^2$$

s.t. $\quad \text{CBC}(x, u) \geq 0.$

(5)

In practice, the QPs above cannot be solved infinitely fast. Optimization is typically performed at triggering times $t_k, t_{k+1}, \ldots$ providing control input $u_k \triangleq \pi(x_k)$ when the system state is $x_k \triangleq x(t_k)$. Ames et al. [21, Thm. 3] show that if $f, g$, and $\alpha \circ h$ are locally Lipschitz, then $\pi(x)$ and $\text{CBC}(x, \pi(x))$ 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 $F(x) \in \mathbb{R}^{n \times (1+m)}$ are unknown. We place a prior distribution over $\text{vec}(F(x))$ using a GP [59] with mean function $\text{vec}(M_0(x))$ and covariance function $K_0(x, x')$.

Our objective is to compute the posterior distribution of $\text{vec}(F(x))$ 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, $\text{vec}(F(x)) \sim \mathcal{GP}(\text{vec}(M_0(x)), K_0(x, x'))$, and a training set, $X_{1:k} \triangleq \{x(t_1), \ldots, x(t_k)\}$, $U_{1:k} \triangleq \{u(t_1), \ldots, u(t_k)\}$, compute the posterior distribution $\mathcal{GP}(\text{vec}(M_k(x)), K_k(x, x'))$ of $\text{vec}(F(x))$ at time $t_k$, choose a control input $u_k$ and triggering period $\tau_k$ such that for $u(t) \triangleq u_k$:

$$\mathbb{P}(\text{CBC}(x(t), u_k) \geq 0) \geq p_k \quad \text{for all} \quad t \in [t_k, t_{k+1}),$$

(6)

where $x(t)$ 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 ($X_{1:k}, U_{1:k}, X_{1:k}$). To simplify the derivation, we restrict $x_k \in X_{1:k}$ and $U_{1:k}$ are observed without noise, but each measurement $x(t_k)$ in $X_{1:k}$ is corrupted by zero-mean Gaussian noise $\mathcal{N}(0, S)$ that is independent across time steps. Treating $f(x) + g(x)u$ as a single vector-valued function with input $[x^\top, u^\top]^\top$ 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 $u$. Hence, we focus on learning the matrix-valued function $F(x) = [f(x), g(x)]$ defined in (1).

The simplest approach is to use $n(1+m)$ decoupled GPs for each element of $F(x)$. This approach ignores the dependencies among the components of $f(x)$ and $g(x)$. Furthermore, since the outputs of $f(x)$ and $g(x)$ are observed together via $X_{1:k}$, training data dimensions are still mutually correlated and cannot be treated as decoupled GPs denying the efficiency advantage. At the other extreme, treating $\text{vec}(F(x))$ 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 $K_0(x, x')$ is challenging. A promising approach is offered by the Coregionalization GP (CoGP) model [65], where the kernel function is decomposed as $K_0(x, x') = \Sigma \kappa_0(x, x')$ into a scalar kernel $\kappa_0(x, x')$ and a covariance matrix parameter $\Sigma \in \mathbb{R}^{n(1+m) \times (1+m)}$. Estimating $\Sigma$ may

1If not available, the derivatives may be approximated via finite differences, e.g., $X_{1:k} \triangleq \{x(t_1) - x(t_1), \ldots, x(t_{k+1}) - x(t_k)\}$, 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 $K_0(x, x')$ inspired by the Matrix Variate Gaussian distribution [66], [67]. The strengths of our factorization are that it models the correlation among the elements of $F(x)$, 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 (MGV) 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:

$$p(\mathbf{X}; \mathbf{M}, \mathbf{A}, \mathbf{B}) \propto \exp \left( -\frac{1}{2} \text{tr} \left[ \mathbf{B}^{-1} (\mathbf{X} - \mathbf{M})^\top \mathbf{A}^{-1} (\mathbf{X} - \mathbf{M}) \right] \right)$$

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 MGV 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:

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

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

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 $\mathbf{B}_{1:k} \in \mathbb{R}^{k \times k}$ be a matrix with $k \times k$ block elements $\mathbf{B}_{1:k} \in \mathbb{R}^{k \times k}$ and define $\mathbf{M}_{1:k} \in \mathbb{R}^{k \times k}$, $\mathbf{B}_{1:k} \in \mathbb{R}^{k \times k}$, and $\mathbf{U}_{1:k} \in \mathbb{R}^{k \times k}$. Consider an arbitrary test point $x_s$. The train and test data are jointly Gaussian:

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

where

$$\mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k}^\top = \begin{bmatrix} \mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k} & \cdots & \mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k} \\ \vdots & \ddots & \vdots \\ \mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k} & \cdots & \mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k} \end{bmatrix}$$

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

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

$$\begin{align*}
\mathbf{M}_k(x) \triangleq & \mathbf{M}_0(x) + \left( \mathbf{X}_{1:k} - \mathbf{M}_{1:k} \mathbf{U}_{1:k} \right) (\mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k}^\top)^{-1} \\
\mathbf{B}_k(x, x') \triangleq & \mathbf{B}_0(x, x') - \mathbf{B}_{1:k}(x) \mathbf{U}_{1:k} (\mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k}^\top)^{-1} \\
(\mathbf{U}_{1:k} \mathbf{B}_{1:k}(x))^\top \triangleq & \left( \mathbf{U}_{1:k} \mathbf{B}_{1:k} \mathbf{U}_{1:k} + \sigma^2 \mathbf{I}_k \right)^{-1} \mathbf{U}_{1:k} \mathbf{B}_{1:k}(x). 
\end{align*}$$

Proof. See Appendix B.

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

$$F(x)u \sim \mathcal{G}\mathcal{P}(\mathbf{M}_k(x)u, \mathbf{U}_{1:k} \mathbf{B}_k(x, x')u \otimes \mathbf{A}).$$

A key property of the MVGP model in Prop. 3 is that the posterior preserves the kernel decomposition. In the special case of, $\mathbf{B}_0(x, x') \triangleq \mathbf{K}_0(x, x')$, 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 $(1+m)^2n^2$ parameters for the full covariance decomposition $\mathbf{K}_0(x, x') \triangleq \mathbf{K}_0(x, x')$. 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 $\mathbf{B}_0(x, x') = \mathbf{K}_0(x, x')$) and from a CoGP model. The shapes of the MVGP covariances for different columns of $F(x)$ 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 $F(x)u$ 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 $F(x)u$. We focus on handling probabilistic safety constraints of the form $\mathbb{P}(\mathbf{CBC}(x, u) \geq 0) \geq p$. Since the control Lyapunov and control barrier conditions have the same form, involving the Lie derivative of a known function $V(x)$ or $h(x)$ 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 $t_k, t_{k+1}, \ldots$, at which the system inputs $u$ should be recomputed in order to guarantee safety with high probability along the continuous-time system trajectory, instead of instantaneously in time.
Our approach requires the stochastic system dynamics $F(x)u$ to be locally Lipschitz continuous with high probability. This is necessary to ensure that for any MVGP sample $\tilde{F}(x)$, the ordinary differential equation $\dot{x} = \tilde{F}(x)u$ 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 $f(x) \triangleq F(x)u$ in (11) with distribution $\mathcal{GP}(\mu(x), \kappa_B(x, x'))$, where $\mu(x) = M_e(x)u$ and $\kappa_B(x, x') = u^T B_i(x, x')u$. Suppose that $\kappa_B(x, x')$ has continuous partial derivatives of fourth order. For $j \in \{1, \ldots, n\}$, let $\kappa_{X}^{i,j} \triangleq \frac{\partial^4 \kappa_B(x, x')}{\partial x_i \partial x_j}$ with Lipschitz constant $L_{k}^{i,j}$ over a compact set $\mathcal{X}$ with maximal extension $\tau = \max_{x, x' \in \mathcal{X}} \|x - x'\|$. Then, each element of the Jacobian of $f(x)$ is bounded with probability of at least $1 - \frac{2}{n^2}$:

$$\left| \sup_{x \in \mathcal{X}} \frac{\partial F(x)}{\partial x_j} \right| \leq L_{i,j}^{i,j}, \quad \forall i, j \in \{1, \ldots, n\},$$

where

$$L_{i,j}^{i,j} \triangleq 2 \log \left( \frac{2n^2}{\delta_L} \right) \kappa_{X}^{i,j} + 12 \sqrt{n} \max \left\{ \kappa_{X}^{i,j}, \sqrt{\tau A_i L_{k}^{i,j}} \right\}$$

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

**Proof.** See Appendix D. \qed

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 $F(x)u$ at time $t_k$ in (11):

$$\pi(x_k) \in \arg \min_{u_k} \|R(x_k)u_k\|^2 + \lambda \delta^2$$

s.t. $P(\text{CLF}(x_k; u_k) \leq \delta \mid x_k, u_k) \geq \tilde{p}_k$

$$P(\text{CBC}(x_k; u_k) \geq \zeta \mid x_k, u_k) \geq \tilde{p}_k,$$

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

**Remark 1.** Given a desired safety probability $p_k$ for the interval $[\tilde{t}_k, t_k + \tau_k]$, 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 $\delta_L$ affects the Lipschitz constant $L^2$ and in turn the duration $\tau_k$, 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 $\tau_k$ for which it remains valid. The choice of $\zeta$ and its effect on $\tau_k$ 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 $x_k$ and $u_k$, $CBC_k \triangleq CBC(x_k, u_k)$ is a Gaussian random variable with the following parameters:

$$E[CBC] = \nabla_x h(x_k)^	op M_k(x_k) u_k + \alpha(h(x_k)), \quad (14)$$

$$\text{Var}[CBC] = (u_k^\top B_k(x_k, x_k) u_k) \nabla^2_h x h(x_k) \mathbf{A} \nabla x h(x_k)), \quad (15)$$

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

$$CBC_k = \nabla x h(x_k)^	op F(x_k) u_k + \alpha(h(x_k)). \quad (16)$$

Then, the mean and variance of CBC are:

$$E[CBC] = \nabla_x h(x_k)^	op E[F(x_k) u_k] + \alpha(h(x_k)), \quad (17)$$

$$\text{Var}[CBC] = \nabla_x h(x_k)^	op \text{Var}[F(x_k) u_k] \nabla x h(x_k), \quad (18)$$

where the mean and variance of $F(x_k) u_k$ are from (11). \hfill \square

Using Lemma 2, we can rewrite the safety constraint as

$$P(CBC_k \geq \zeta | x_k, u_k) = 1 - \Phi \left( \frac{\zeta - E[CBC]}{\text{Var}[CBC]} \right) \geq \tilde{p}_k, \quad (19)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian. Note that if the control input is chosen so that $\frac{\zeta - E[CBC]}{\text{Var}[CBC]} \rightarrow -\infty$, as the posterior variance of $CBC_k$ tends to zero, the probability $P(CBC_k \geq \zeta | x_k, u_k)$ 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 (19) and noting that $\Phi^{-1}(1 - \tilde{p}_k) = -\sqrt{2} \text{erf}^{-1}(2\tilde{p} - 1)$, where erf(·) is the Gauss error function, the optimization problem in (13) can be restated as:

$$\pi(x_k) \in \operatorname*{arg\,min}_{u_k \in U, \delta \in \mathbb{R}} \left\| R(x_k) u_k \right\|^2 + \lambda \delta^2$$

$$\text{s.t. } \delta - E[\text{CLC}_k] \geq \frac{c(\tilde{p}_k) / \text{Var}[\text{CLC}_k]}{\sqrt{2} \text{erf}^{-1}(2\tilde{p}_k - 1)} \geq \frac{\zeta - E[CBC]}{\text{Var}[CBC]}, \quad (20)$$

where $c(\tilde{p}_k) \triangleq \sqrt{2} \text{erf}^{-1}(2\tilde{p}_k - 1)$. Using the reformulation in (20), 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):

$$\pi(x_k) \in \operatorname*{arg\,min}_{y \in \mathbb{R}, \delta \in \mathbb{R}, u \in U} y$$

$$\text{s.t. } y - \left\| R(x_k) u + \sqrt{\delta} \right\|^2 \geq 0$$

$$q_k(x_k) u - \delta + c(\tilde{p}_k) \text{Var}(\text{CLC}_k) \leq 0$$

$$e_k(x_k) u - \zeta - c(\tilde{p}_k) \text{Var}(\text{CBC}_k) \geq 0, \quad (21)$$

where $q_k(x_k) \triangleq M_k(x_k) \nabla x V(x_k) + \left[ \gamma(V(x_k)) \right]_0$, $P_k(x_k) \triangleq \nabla x V(x_k) \mathbf{A} \nabla x V(x_k) B_k^2(x_k, x_k)$, $e_k(x_k) \triangleq M_k(x_k) \nabla x h(x_k) + \left[ \alpha(h(x_k)) \right]_0$, $V_k(x_k) \triangleq \nabla x h(x_k) \mathbf{A} \nabla x h(x_k) B_k^2(x_k, x_k)$, and $B_k^2(x_k, x_k)$ is the Cholesky factorization of $B_k(x_k, x_k)$.

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. \hfill \square

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 $\epsilon$ 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 Castañeda et al [73] for discussion of feasibility conditions of an SOCP controller. \hfill \checkmark

### 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 $L_{x_k}$, then for all $s \in [0, \tau_k)$ the total change in the system state is bounded with probability at least $1 - \delta_L$:

$$\| x(t_k + s) - x_k \| \leq \tau_k(s) \triangleq \frac{1}{L_{x_k}} \| x_k \| \left( e^{L_{x_k} t_k} - 1 \right). \quad (22)$$
Recall from Sec. III-B that \( h \) is a continuously differentiable function. Thus, using Lemma 3, for any inter-triggering time \( \tau_k \), there exist a constant \( L_{h_k} > 0 \) such that:

\[
\sup_{s \in [0, \tau_k]} \| \nabla_x h(x(t_k + s)) \| \leq L_{h_k}. \tag{21}
\]

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 \( C \). Assume the system dynamics have Lipschitz constant \( L_f \) with probability at least \( 1 - \delta_L \) (Lemma 1), the system is safe with margin \( \zeta \) at triggering time \( t_k \) with probability at least \( \bar{p}_k = p_k/(1 - \delta_L) \) (Prop. 4), and \( \alpha \) is Lipschitz continuous with Lipschitz constant \( L_{\alpha} \). Then, the system is safe according to the constraint in (6) with probability at least \( \bar{p}_k \) for time duration \( \tau_k \leq \frac{1}{L_f} \ln \left( 1 + \frac{L_h}{L_{h_k} \cdot L_{\alpha} \cdot L_{f_k}} \right) \), where \( L_{h_k} \) is defined in (21).

**Proof.** Since the program (13) has a solution at the triggering time \( t_k \), we know \( \mathbb{P}(\text{CBC} \geq \zeta | x_k, u_k) \geq \bar{p}_k \). Thus, conditioned on CBC \( \geq \zeta \) and the Lipschitz continuity of the system dynamics, if we prove 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 \( \alpha \) and \( h \) (from (21)), for all \( s \in [0, \tau_k] \), we have with probability at least \( \bar{p}_k = 1 - \delta_L \):

\[
|\text{CBC}(x(s + t_k), u_k) - \text{CBC}(x_k)| \leq \left( \sup_{s \in [0, \tau_k]} \| \nabla_x h(x(s + t_k)) \| L_{h_k} + L_{\alpha} L_{h_k} \right) \tau_k(s) \tag{22}
\]

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

Prop. 4 ensures instantaneous safety with probability \( \bar{p}_k \geq p_k \), 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 \( \mathbb{P}(\text{CBC} \geq \zeta) \geq p_k \) for all \( t \in [t_k, t_k + \tau_k] \), due to Prop. 2. Prop. 5 characterizes the longest time \( \tau_k \) until it is necessary to recompute the control input to guarantee safety.

Fig. 2 shows an evaluation of the Lipschitz constant \( L_{t_k} \) (Lemma 1) and triggering time \( \tau_k \) (Prop. 5) for a system with Ackermann dynamics. We compute \( L_{t_k} \) and \( \tau_k \) 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 degree\(^2\), using 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 \( h(x) \), i.e., \( L_g L_f^{(r-1)} h(x) \neq 0 \) and \( L_g L_f^{(r-1)} h(x) = 0, \forall k \in \{1, \ldots, r-1\} \). This condition can be expressed by defining

\[
\eta(x) = \frac{h(x)}{L_f h(x)} + \mathcal{F}, \quad \mathcal{F} = \begin{bmatrix} 0 & \ldots & 0 \\ 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \\ 0 & \ldots & 0 \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},
\]

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

\[
\eta(x) = \mathcal{F} \eta(x) + \mathcal{G} u, \quad h(x) = c^T \eta(x),
\]

where \( c \in [1, 0, \ldots, 0]^T \in \mathbb{R}^r \).

**Definition 4** [25], [37]. A function \( h \in C^r(\mathcal{X}, \mathbb{R}) \) is an exponential control barrier function (ECBF) for the system in (1) if there exists a vector \( k_\alpha \in \mathbb{R}^r \) such that the \( r \)-th order control barrier condition \( \text{ECBF}^{(r)}(x, u) \triangleq \mathcal{L}^{(r)} f(x) + \mathcal{L}_g L_f^{(r-1)} h(x) u + k_\alpha \eta(x) \) satisfies

\[
\sup_{x \in \mathcal{X}} \text{ECBF}^{(r)}(x, u) \geq 0, \quad \forall x \in \mathcal{X}
\]

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

B. Probabilistic EECF Constraints

As in (13), we are interested in solving

\[
\pi(x_k) \in \arg \min_{u_k \in U} \mathbb{E}[|R(x_k) u_k|^2]
\]

s.t. \( \mathbb{P}(|\text{CBC}^{(r)}_k| \geq \zeta | x_k, u_k) \geq \bar{p}_k \),

where \( \pi(x_k) \) is the control input to the system at time \( k \), \( \text{CBC}^{(r)}_k \) is the control barrier condition at time \( k \), \( \zeta \) is a safety margin, and \( \bar{p}_k \) is the safety probability. The problem is to find control inputs that ensure the system remains within the safe set with a given probability for all future times.

Fig. 2. Lipschitz constant \( L_{t_k} \) and triggering time \( \tau_k \) 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 differ. See Sec. VIII-B3 for details.
where $\mathrm{CBC}^{(r)}_k \triangleq \mathrm{CBC}^{(r)}(x_k, u_k)$ and we have dropped the CLC term for clarity of presentation. While we explicitly characterised the distribution of $\mathrm{CBC}^{(r)}_k$ for relative degree one in Lemma 2, the distribution of $\mathrm{CBC}^{(r)}_k$ cannot be determined explicitly for arbitrary $r$. Instead, we use a concentration bound to reweight the chance constraint in terms of the moments of $\mathrm{CBC}^{(r)}_k$. Before we can derive a SOCP controller for higher relative-degree systems, we need to show that the mean and variance of $\mathrm{CBC}^{(r)}_k$ are to rewrite the chance constraint in terms of the moments of $\mathrm{CBC}^{(r)}_k$. While we explicitly explicitly for arbitrary $(\text{Lemma 1})$ and (Proposition 7 characterizes the longest time

time $t_k$. Given $h(x(t)) \geq 0$, we find an upper bound on the sampling time $\tau_k$ that ensures $h(x(t))$ 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 $\mathcal{C} = \{ x \in \mathbb{X} | h(x) \geq 0 \}$. Suppose the system has relative degree $r \geq 1$ with respect to $h(x)$ and the SOCP in (28) has a solution at triggering time $t_k$ with $h(x_k) \triangleq h(x) - \zeta_b \geq 0$ as the CBF and $\zeta = 0$. Suppose that the system dynamics are Lipschitz continuous with probability at least $1 - \delta_k$ (Lemma 1) and $h$ is Lipschitz continuous with Lipschitz constant $L_{h_k}$. Then, $h(x(t)) \geq 0$, for $t \in [t_k, t_k + \tau_k]$, holds with probability $\tilde{p}_k = \tilde{p}_k(1 - \delta_k)$, where $\tau_k \leq \frac{1}{L_{h_k}} \ln \left(1 + \frac{L_{h_k} \zeta_b}{\|x_k\|} \right)$.

**Proof.** Using Lemma 1 and 3 we have,

$$
\sup_{s \in [0, \tau_k]} |h(x(t_k + s)) - h(x_k)| \leq L_{h_k} \|x_k + s - x_k\|
$$

$$
\leq \frac{L_{h_k}}{L_{t_k}} \|x_k\| (e^{L_{t_k} \tau_k} - 1) \leq \zeta_b,
$$

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

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

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

**VIII. 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:

$$
\text{CBC}^{(2)}(x, u) = [\nabla_x \mathcal{L} h(x)]^\top F(x) u + [h(x), \mathcal{L}_f h(x)]^\top k_v,
$$

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

**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 temporally introduce a modified CBF $h_{\text{mod}}(x) \triangleq h(x) - \zeta_b$ with $\zeta_b > 0$. We solve the SOCP in (28) with $\zeta = 0$ but replace $h(x)$ with $h_{\text{mod}}(x)$. Enforcing the safety constraint in (28) for $h_{\text{mod}}(x_k)$, ensures that with probability $\tilde{p}_k$, $h(x_k) \geq \zeta_b$ at the sampling times. We find an upper bound on the sampling time $\tau_k$ that ensures $h(x(t))$ 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 $t_k$.

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

$$
\begin{align*}
\mathbb{E}[\mathrm{CBC}^{(r)}(x, u)] &= e^{(r)}(x)^\top u, \\
\text{Var}[\mathrm{CBC}^{(r)}(x, u)] &= u^\top V^{(r)}(x) u, \\
\end{align*}
$$

where

$$
\begin{align*}
e^{(r)}(x) &\triangleq \mathbb{E} \left[ F(x)^\top \nabla_x \mathcal{L}^{(r-1)}_f h(x) + \begin{bmatrix} k_r \eta(x) \\ 0_m \end{bmatrix} \right], \\
V^{(r)}(x) &\triangleq \mathbb{V}[ F(x)^\top \nabla_x \mathcal{L}^{(r-1)}_f h(x) + \begin{bmatrix} k_r \eta(x) \\ 0_m \end{bmatrix} ]^{1/2}.
\end{align*}
$$

**Proof.** See Appendix E.

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 $h(x) \in \mathbb{C}^r(\mathcal{X}, \mathbb{R})$. If the system inputs are determined in continuous time $t_k$ from the following SOCP,

$$
\begin{align*}
\pi(x_k) &= \arg\min_{y \in \mathbb{R}, \xi, u_k} y \\
&\text{s.t.} \quad y - \|R(x_k) u_k\| \geq 0 \\
e^{(r)}_k(x_k)^\top u_k - \zeta - c^{(r)}(\tilde{p}_k) \|V^{(r)}_k(x_k) u_k\| &\geq 0,
\end{align*}
$$

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 $\mathcal{C} = \{ x \in \mathcal{X} | h(x) \geq 0 \}$ with probability $\tilde{p}_k$.

**Proof.** See Appendix F.

In the proposition above, we bound $\mathbb{P}(\mathrm{CBC}^{(r)}_k \geq \zeta_k | x_k, u_k)$ in (24) using Cantelli’s inequality, because the exact distribution of $\mathrm{CBC}^{(r)}_k$ is unknown. The safety constraint in (28) can be interpreted as: “The system must satisfy $\mathrm{CBC}^{(r)}_k$ in expectation by a margin $c(\tilde{p}_k)$ times the standard deviation of $\mathrm{CBC}^{(r)}_k$”. Solving the SOCP requires knowledge of the mean and variance of $\mathrm{CBC}^{(r)}_k$. Sec. VII shows how to determine the mean and variance of $\mathrm{CBC}^{(r)}_k$, which is of practical relevance for many physical systems. For $r > 2$, Monte Carlo sampling can be used to estimate these quantities.
shows a computation graph for CBC(2)(x, 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 x, y, z with means \( \bar{x}, \bar{y}, \bar{z} \) and variances \( \text{Var}[x], \text{Var}[y], \text{Var}[z] \), respectively. Let their pairwise covariances be \( \text{cov}(x, y), \text{cov}(y, z) \) and \( \text{cov}(z, x) \). The mean, variance, and covariances of \( x' y \) are given by:

\[
\begin{align*}
\mathbb{E}[x' y] &= x' y + \frac{1}{2} \text{tr}(\text{cov}(x, y) + \text{cov}(y, x)) \\
\text{Var}[x' y] &= \frac{1}{2} \text{tr}(\text{cov}(x, y) + \text{cov}(y, x))^2 + y' \text{Var}[y] y + x' \text{Var}[x] x + x' \text{cov}(y, x) y + y' \text{cov}(x, y) y + \text{cov}(x, y') x + \text{cov}(y, x') y \\
\text{cov}(x', y') &= \begin{bmatrix} \text{cov}(x, y) & \text{cov}(y, x) \\ \text{cov}(y, x) & \text{cov}(x, y) \end{bmatrix} + \begin{bmatrix} \text{var}(x' x) & \text{var}(x' y) \\ \text{var}(y' x) & \text{var}(y' y) \end{bmatrix}.
\end{align*}
\]

**Proof.** See Appendix G.

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

\[
\text{cov}(\nabla_x q(x), s(x')) = \nabla_x \text{cov}_{q,s}(x, x').
\]

The mean and variance of CBC(2)(x, u) can be computed using Lemma 4:

\[
\begin{align*}
\mathbb{E}[\text{CBC}(2)(x, u)] &= \mathbf{e}^{(2)}(x)' \mathbf{u} \\
\text{Var}[\text{CBC}(2)(x, u)] &= \mathbf{u}' \mathbf{V}^{(2)}(x) \mathbf{V}^{(2)}(x)' \mathbf{u} \\
\mathbf{e}^{(2)}(x) &= \mathbb{E}[F(x)'] \nabla_x \mathcal{L}_f h(x) \\
&\quad + \begin{bmatrix} k_{a,1} h(x) + k_{a,2} \mathbb{E}[\mathcal{L}_f h(x)] \\ 0_m \end{bmatrix} \\
\mathbf{V}^{(2)}(x) &= \begin{bmatrix} \text{Var}[F(x)'] & \nabla_x \mathcal{L}_f h(x) \end{bmatrix}
\end{align*}
\]

**Algorithm 1:** Mean and variance of CBC(2)(x, u).

**Data:** ECFB \( h(x) \), system dynamics distribution \( \mathcal{G}(F(x)) \), system parameters \( \mathbf{M}_k(x), \mathbf{B}_k(x) \). Let \( \mathcal{GP}(\mathbb{E}[x], \mathbf{V}(x)) \otimes \mathbf{A} \)

**Result:** \( \mathcal{GP}(\mathbb{E}[x], \mathbf{V}(x)) \otimes \mathbf{A} \)

1. Compute \( \mathbb{E}[\mathcal{L}_f h(x)] = \nabla_x h(x) \mathcal{M}_k(x) \mathbf{1}_m \)
2. Compute the mean and variance of \( \nabla_x \mathcal{L}_f h(x) \) using Lemma 5.
3. Compute \( \text{cov}(\nabla_x \mathcal{L}_f h(x), \mathcal{L}_f h(x)) \) using (33)
4. Compute the mean and variance of \( \mathcal{L}_f h(x) \) using Lemma 5.
5. Compute \( \mathcal{L}_f h(x) \)
6. Plug the above values into (34) to obtain \( \mathcal{G}(F(x)) \).

**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 Github.

**A. Pendulum**

Consider a pendulum, shown in Fig 6, with state \( x = [\theta, \dot{\theta}, \omega]^T \), containing its angular deviation \( \theta \) from the rest position and its angular velocity \( \omega \). The pendulum dynamics are:

\[
\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{g}{\ell} \sin(\theta) + \frac{\ell}{m} \sin(\theta) \\ \frac{1}{m} \end{bmatrix} u_\ell
\]

where \( g \) is the gravitational acceleration, \( m \) is the mass, and \( \ell \) is the length.

1) Estimating Pendulum Dynamics: We compare our MVGP model versus the Coregionalization GP (CoGP) 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.

We let \( \mathbf{K}_0(x, x') = \mathbf{B}_0(x, x') \otimes \mathbf{A} \) for MVGP and \( \mathbf{K}_0(x, x') = \Sigma_0(x, x') \) for CoGP; where \( \Sigma_0(x, x') \) is a scaled radial-basis function kernel. Further, \( \mathbf{A}, \mathbf{B}, \) and \( \Sigma \) are constrained to be positive semi-definite by modeling each as \( \Sigma_0 \in \mathbb{R}^{m \times m} \).
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 \( f(x) \) and \( g(x) \). We approximate decoupled GP inference by constraining the matrices \( A, B, \) and \( \Sigma \) 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 \( T \), we compute variance-weighted error as,

\[
\text{err}(T) \triangleq \sqrt{\frac{1}{|T|} \sum_{x \in T} \|K_p^{-1}(x, x) \text{vec}(M_k(x) - F(x))\|_2^2}.
\]  

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™ 2080 Ti 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 \( \theta_c - \Delta_{\text{rad}}, \theta_c + \Delta_{\text{rad}} \) with \( \theta_c = 45^\circ, \Delta_{\text{rad}} = 22.5^\circ \) that needs to be avoided, as shown in Fig. 6. The control barrier function defining this safe set is \( h(x) = \cos(\Delta_{\text{rad}}) - \cos(\theta - \theta_c) \). 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 \( M_0(x) = 0_{n \times (1+m)} \) and randomly generated covariance matrices \( A \) and \( B \) are used to initialize the MVGP model. We formulate a SOC as in (28) with \( r = 2 \). We specify a task requiring the pendulum to track a reference control signal \( \hat{x}_k(x) \) and specify the optimization objective as \( \|R(x_k)(u_k - \hat{x}_k(x_k))\|_2^2 \) with \( R(x_k) = I \). The reference signal \( \hat{x}_k(x) \) is an \( \varepsilon \)-greedy policy \([82]\), used to provide sufficient excitation in the training data. Concretely, \( \hat{x}_k(x) \) is sampled from an \( \varepsilon_k \) weighted combination of Dirac delta distribution \( \delta \) and Uniform distribution \( U \), \( \hat{x}_k(x_k) \sim (1 - \varepsilon_k) \delta_{u=0} + \varepsilon_k U[-20, 20] \), where \( \varepsilon_k = 10^{-k/30} \) so that \( \varepsilon_k \) 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 SOC 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 \( x = [x, y, \theta] \), including position \( (x, y) \) and orientation \( \theta \). To make the Ackermann dynamics affine in the control input, we define

\[ z \triangleq v \tan(\phi), \]  

where \( v \) is the linear velocity and \( \phi \) is the steering angle. The Ackermann-drive dynamics are:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \quad 1
\end{bmatrix}
\begin{bmatrix}
\cos(\theta) \\
\sin(\theta) \\
0
\end{bmatrix}
\begin{bmatrix}
v \\
\phi
\end{bmatrix},
\]

(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., \( F(x) = F([0, 0, \theta]') \), \( \forall x \in \mathbb{R}^3 \). 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 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 \( c_i \in \mathbb{R}^2 \) and radius \( r_i > 0 \) is:

\[ h_i(x) = q_1(\|x_{1:2} - c_i\|_2^2 - r_i^2) + q_2 \cos(\theta - \phi_i), \]  

(38)

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

\[ V(x, x^{(d)}) = \frac{w_1}{2} \rho^2 + w_2(1 - \cos(\alpha)), \]  

(39)

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

\[
\begin{aligned}
&\text{CLC}(x, u) = \nabla_x^\top V(x, x^{(d)}) F(x) u + \\
&\nabla_{x^{(d)}}^\top V(x, x^{(d)}) x^{(d)} + \gamma_u V(x, x^{(d)}).
\end{aligned}
\]  

(40)

The control Barrier condition for each obstacle \( i \) is:

\[ \text{CBC}(x, u; i) = \nabla_x^\top h_i(x) F(x) u + \gamma_{hi} h_i(x). \]  

(41)

We implement the SOC controller in Prop. 4 with parameters:

\[ q_1 = 0.7, q_2 = 0.3, w_1 = 0.9, w_2 = 1.5, \gamma_{hi} = 5, \gamma_V = 10, \tau = 0.01. \]

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 \( c(\hat{p}_k) = 0.5 = 0 \). In this case, the SOC in (19) reduces to the deterministic-case QP in (4). We dub the case of \( \hat{p}_k = 0.5 \) as Mean CBF and the case of \( \hat{p}_k = 0.99 \) as Bayes CBF. In both the cases, the covariance matrices are initialized as \( A = 0.011I_3 \) and \( B = I_3 \). The true dynamics, \( F(x) \), is specified with \( L = 12 \), while the
mean dynamics $M_0(x)$ is obtained with $L = 1$. The result 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 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 $\delta_k = 0.99$. The true dynamics are specified with $L = 1$, while prior mean dynamics correspond to $L = 8$. The covariance matrices $A$ and $B$ are initialized with elements independently sampled from a standard Gaussian distribution while ensuring them to 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 $L_\delta$ (Lemma 1) and triggering time $\tau_k$ (Prop 5) for the Ackermann vehicle. At any time $k$, we use the following choice of parameters to compute $\tau_k$, $\delta_L = 10^{-4}$, $\zeta = 10^{-2}$, $L_{\alpha} = 1$, and $L_{h_k} = \max_{x \in X_k} \nabla h(x)$. We define $X_k$ to be a cuboid with sides $(0.2, 0.2, \frac{\pi}{2})$ in the directions $x, y$ and $\theta$ respectively centered around the current state $x_k$. We compute the Lipschitz constant $L_\delta$ and $\tau_k$ both numerically.

![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 $O(k^3 t^3)$), 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 $O(k^3 n^2)$, 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.]

![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.]

and analytically. The numerical computation of $L_{ij}$ is done by sampling the GP $\frac{\partial E(x)}{\partial x}$ 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 $\tau_k$ is the same as the analytical $\tau_k$. 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 $\delta_L$ is determined by the samples drawn from the GP. We also observe that the second term in the expression for $L_{ij}$ (Lemma 1) dominates the effect of $\delta_L$ 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 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 $X$ follow an MVG distribution $\mathcal{M}\mathcal{N}(M, A, B)$. Then, $\text{vec}(X) \sim \mathcal{N}(\text{vec}(M), B \otimes A)$.

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

$$CX \sim \mathcal{M}\mathcal{N}(CM, CAC^T, B),$$

$$XD \sim \mathcal{M}\mathcal{N}(MD, A, D^T BD).$$

APPENDIX B

PROOF OF PROPOSITION 3

Let $\mathcal{G}P(\text{vec}(M_k(x)), K_k(x, x'))$ be the posterior distribution of $\text{vec}(F(x))$ conditioned on the training data $(X_{1:k}, U_{1:k}, X_{1:k})$. The mean and variance can be obtained by applying Schur’s complement to (8),

$$\text{vec}(M_k(x)) = \text{vec}(M_0(x)) +$$

$$(B_{1:k} \otimes A)^T (B_{1:k} \otimes A)^{-1} \text{vec}(X - \mathcal{M}_{1:k}U_{1:k}),$$

$$\text{vec}(V_k(x)) = \text{vec}(V_0(x)) +$$

$$(B_{1:k} \otimes A)^T (B_{1:k} \otimes A)^{-1} \text{vec}(X - \mathcal{M}_{1:k}U_{1:k}),$$

$$\text{vec}(V_k(x)) = \text{vec}(V_0(x)) +$$

$$(B_{1:k} \otimes A)^T (B_{1:k} \otimes A)^{-1} \text{vec}(X - \mathcal{M}_{1:k}U_{1:k}),$$
\( K_k(x, x) = B_0(x, x) \otimes A - \left( B_{1:k}(x) \otimes A \right) \left( (B_{1:k}(x) \otimes A)^{-1} (B_{1:k}(x) \otimes A) \right). \)  
(44)

where \( B_{1:k} \triangleq U_{1:k}^T B_{1:1}U_{1:k} + \sigma^2 I_k \) and \( B_{1:k}(x) \triangleq B_{1:k}(x)U_{1:k}. \) For appropriately sized matrices \( P, Q, R, S, \) the Kronecker product satisfies \((P \otimes Q)(R \otimes S) = (PR \otimes QS)\) and \((P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1}. \) Thus, we can rewrite the mean as:

\[
\text{vec}(M_k(x)) = \text{vec}(M_0(x)) + \left( (B_{1:k}(x)^{-1} \otimes A) A^{-1} \right) \text{vec}(\hat{X} - M_{1:k}U_{1:k}).
\]
(45)

Applying \((P \otimes Q)\vec{v}(R) = \vec{v}(QR^T),\) we get

\[
M_k(x) = M_0(x) + (\kappa_{1:k} M_{1:k} U_{1:k}) (U_{1:k}^{-1} B_{1:k}(x))^{-1} U_{1:k}^T (x).
\]
(46)

Similarly, the covariance can be rewritten as,

\[
K_k(x, x') = \left( B_0(x, x') - B_{1:k}(x')U_{1:k}^{-1} B_{1:k}^T(x') \right) \otimes A.
\]
(47)

Defining \( B_k(x, x') \) such that \( K_k(x, x') = B_k(x, x') \otimes A, \) we can write the posterior distribution of \( \text{vec}(F(x)) \) as \( \mathcal{GP}(\text{vec}(M_k(x)), B_k(x, x') \otimes A). \)

\[\square\]

**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 \( \text{vec}(F(x)) \sim \mathcal{GP}(\text{vec}(M_0(x)), \Sigma_{\kappa_0}(x, x')) \), where \( \Sigma \in \mathbb{R}^{(1+m)x(1+m)n} \) is the covariance matrix of the output dimensions, and \( \kappa_0(x, x') \) is the kernel function. Denote the kernel function evaluation over the training data as,

\[
K_{1:k}^T \triangleq \begin{bmatrix}
   \kappa_0(x_1, x_1) & \ldots & \kappa_0(x_k, x_1) \\
   \vdots & \ddots & \vdots \\
   \kappa_0(x_k, x_1) & \ldots & \kappa_0(x_k, x_k)
\end{bmatrix} \in \mathbb{R}^{k \times k}
\]
(48)

\[
K_{1:k}(x) \triangleq \begin{bmatrix}
   \kappa_0(x_1, x) & \ldots & \kappa_0(x_k, x)
\end{bmatrix} \in \mathbb{R}^{1 \times k}
\]

Let \( \mathcal{M}_{1:k} \) and \( U_{1:k} \) be defined as in Proposition 3. Then, the posterior distribution of \( \text{vec}(F(x)) \), given the training data \( (X_{1:k}, X_{1:k}, U_{1:k}) \), is \( \mathcal{GP}(\text{vec}(M_k(x)), K_k(x, x')) \), where

\[
\text{vec}(M_k(x)) \triangleq \text{vec}(M_0(x)) + \kappa_0(x, x') \text{vec}(\hat{X}_{1:k} - \mathcal{M}_{1:k} U_{1:k}),
\]
(49)

\[
K_k(x, x') \triangleq \Sigma_{\kappa_0}(x, x') - K_{1:k}(x) K_{1:k}^T (K_{1:k} + \Sigma_{\kappa_0}(x, x') U_{1:k} \otimes I_n)^{-1} K_{1:k}(x),
\]

\[
K_{1:k} \triangleq \begin{bmatrix}
   \mathcal{M}_{1:k} \otimes I_n \\
   U_{1:k} \otimes \Sigma \\
   U_{1:k} \otimes I_n \otimes \Sigma
\end{bmatrix}.
\]

**Proof.** Using \( \text{vec}(PQR) = (RT \otimes P) \text{vec}(Q), \) we can rewrite \( F(x)u \) as

\[
F(x)u = \text{vec}(I_n F(x)u) = (u^T \otimes I_n) \text{vec}(F(x)).
\]
(50)

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

\[
F(x)u \sim \mathcal{N}(M_0(x)u, (u^T \otimes I_n) \Sigma_{\kappa_0}(x, x')(u \otimes I_n)).
\]
(51)

The training data is generated from \( \hat{x} = F(x)u + w \) with \( w \sim \mathcal{N}(0, \Sigma) \) and is jointly Gaussian:

\[
\text{vec}(X_{1:k}) \sim \mathcal{N}(\text{vec}(\mathcal{M}_{1:k}U_{1:k}), K_{1:k}^T).
\]
(52)

The covariance for a single point is given by,

\[
\text{cov}(\text{vec}(F(x)), \text{vec}(F(x')u)) = \Sigma_{\kappa_0}(x, x')(u \otimes I_n).
\]
(53)

A similar expression for covariance can be obtained between the training data and the test data, \( \text{cov}(\text{vec}(F(x)), \hat{X}_{1:k}) = K_{1:k}(x) \).

We can now write the joint distribution between the training and test data,

\[
\text{vec}(F(x)) \sim \mathcal{N} \left( \begin{bmatrix} M_0(x) & 
\kappa_{1:k}^T \end{bmatrix}, \begin{bmatrix} K_{1:k}^T \kappa_{1:k} \end{bmatrix} \right).
\]

\[
\text{vec}(F(x)) \sim \mathcal{N} \left( \begin{bmatrix} M_0(x) & 
\kappa_{1:k} \end{bmatrix}, \begin{bmatrix} K_{1:k} & 
\kappa_{1:k} \end{bmatrix} \right).
\]

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 \( f(x) \) is due to [69, Lemma B.2] with probability at least \( \frac{1}{1 - \frac{2L}{n}}, \)

\[
|f_i(x_1) - f_i(x)| \leq L_{i,j} |x_j - x_j| \quad \forall i, j \in \{1, \ldots, n\}.
\]
(54)

Denote the event in (54) by \( E_i, \dot{E}_j \) so that \( \mathbb{P}(E_i, \dot{E}_j) \geq 1 - \frac{2L}{n}. \) A lower bound on the probability of intersection of all events can be computed using a union bound:

\[
\mathbb{P} \left( \bigcap_{i=1}^n \bigcap_{j=1}^n E_i, \dot{E}_j \right) = 1 - \sum_{i=1}^n \sum_{j=1}^n \mathbb{P}(E_i, \dot{E}_j) \geq 1 - \frac{1}{n} \delta.
\]
(55)

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

\[
n |f_i(x_1) - f_i(x)| \leq \sum_{j=1}^n L_{i,j} |x_j' - x_j| \leq \sqrt{\sum_{j=1}^n L_{i,j}^2} \|x' - x\|_2.
\]

Squaring and adding all inequalities for \( i \in \{1, \ldots, n\} \), we get:

\[
\|f(x') - f(x)\|_2 \leq \sqrt{\sum_{i=1}^n \sum_{j=1}^n L_{i,j}^2} \|x' - x\|_2.
\]

\[\square\]

**APPENDIX E**

**PROOF OF LEMMA 4**

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

\[
\text{CBC}^{(r)}(x, u) = \mathbb{L}^{(r)}(h(x)) + \mathbb{L}_g^{(r-1)}(h(x))u + k_n \eta(x)
\]

\[
= \mathbb{L}_f^{(r)}(h(x)) + \mathbb{L}_g^{(r-1)}(h(x))u + k_n \eta(x)
\]

\[
= \nabla_f \mathbb{L}^{(r-1)}(h(x))u + \nabla_g \mathbb{L}^{(r)}(h(x))u + k_n \eta(x)
\]

\[
= \nabla_f \mathbb{L}^{(r-1)}(h(x))u + \mathbb{L}^{(r)}(h(x))u + k_n \eta(x).
\]
Since, $u = [1, u^\top]^T$, we can rewrite the above as

$$
CBC^{(r)}(x, u) = \left( \nabla_x L_f^{(r-1)}h(x)^\top F(x) + \left[ k_\eta^\top(x) 0_m \right]^\top \right) u
$$

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

$$
\mathbb{E}[CBC^{(r)}(x, u)] = \left( \mathbb{E}[F(x)^\top \nabla_x L_f^{(r-1)}h(x)] + \mathbb{E}\left[ k_\eta^\top(x) 0_m \right] \right) u.
$$

(56)

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

$$
\text{Var}[CBC^{(r)}(x, u)] = \left( \nabla_x L_f^{(r-1)}h(x)^\top F(x) + \left[ k_\eta^\top(x) 0_m \right]^\top \right) u.
$$

(57)

\[ \Box \]

**APPENDIX F**

**PROOF OF PROPOSITION 6**

Unlike CBC for relative degree one, the distribution of $CBC^{(r)}$ 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 $CBC^{(r)}$. For any scalar $\lambda > 0$, we have

$$
\mathbb{P}\left( CBC^{(r)}_k \geq \mathbb{E}[CBC^{(r)}_k] - \lambda \mid x_k, u_k \right) \geq 1 - \frac{\text{Var}[CBC^{(r)}_k]}{\text{Var}[CBC^{(r)}_k] + \lambda^2}.
$$

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.,

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

(58)

The terms can be rearranged into,

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

(59)

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

$$
\mathbb{E}[CBC^{(r)}_k] - \zeta \geq \sqrt{\frac{\tilde{p}_k}{1 - \tilde{p}_k}} \text{Var}[CBC^{(r)}_k].
$$

(60)

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

$$
\pi(x_k) \in \arg \min_{u_k \in \mathcal{U}} \| R(x_k)(x_k)u_k \|
$$

(61)

s.t. $e^{(r)}_k(x_k)^\top u_k - \zeta - c^{(r)}(\tilde{p}_k) \left\| V^{(r)}_k(x_k)u_k \right\|_2 \geq 0,$

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 $\gamma$, leading to (28).

\[ \Box \]

**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 $x$ be a Gaussian random vector with mean $\mu$ and covariance matrix $\Sigma$. Let $\Lambda$ be a symmetric matrix and consider the random variable $x^\top \Lambda x$. The mean of $x^\top \Lambda x$ is

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

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

$$
\kappa_r(x^\top \Lambda x) = 2^{r-1}(r - 1)! \text{tr}(\Lambda \Sigma)^r + r \mu^\top \Lambda (\Sigma \Lambda)^{r-1} \mu.
$$

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

$$
\text{Var}[x^\top \Lambda x] = \kappa_2(x^\top \Lambda x) = 2 \text{tr}(\Lambda \Sigma)^2 + 4 \mu^\top \Lambda \Sigma \Lambda \mu.
$$

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

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

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

$$
x^\top y = \frac{1}{2} \begin{bmatrix} x^\top & y^\top & z^\top \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.
$$

(62)

Applying Lemma 10 to (62), shows that:

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

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

$$
+ x^\top \text{Var}[y|x]x + y^\top \text{cov}(x, y)x + x^\top \text{cov}(y, x)y
$$

$$
= 2 \begin{bmatrix} \text{cov}(x, y, x) \\ \text{cov}(y, x, y) \\ \text{cov}(z, x, y) \\ \text{cov}(z, x, z) \\ \text{cov}(z, y, z) \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ 0 \\ 0 \end{bmatrix}.
$$

\[ \Box \]

**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. Syst., 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. Rakhlín, “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 DeepPC,” in *European Control Conference (ECC)*, 2019, pp. 307–312.

[10] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. 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 for tubes for 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 method,” arXiv preprint arXiv:2004.01142, 2020.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *Robotics: Science and Systems*, 2018.

[17] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Frey, “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. Stuck et al., “MAARS: Machine learning-based Analytics for Automated Rover Systems,” in *Aerospace Conf. IEEE*, 2017, pp. 1–17.

[20] P. Pieland 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 functions 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 Conference on Control Technology and Applications (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 Conference (ECC)*, June 2019, pp. 3420–3431.

[26] 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.

[27] L. Wang, E. A. Theodorou, and M. Egerstedt, “Safe learning of quadrotor dynamics using barrier certificates,” in *IEEE Int Conf Robot Automa.*, (ICRA), 2018, pp. 2460–2465.

[28] 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.

[29] Z. Marvi and B. Kiumarsi, “Safe Off-policy Reinforcement Learning Using Barrier Functions,” in *American Control Conf. (ACC)*. IEEE, 2020, pp. 2176–2181.

[30] A. J. Taylor and A. D. Ames, “Adaptive Safety with Control Barrier Functions,” in *American Control Conf. (ACC)*, 2020, pp. 1399–1405.

[31] B. Lop a 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.

[32] 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 Automa.*, (ICRA), 2019, pp. 1961–1967.

[33] 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.
