

Learning Piecewise Residuals of Control Barrier Functions for Safety of Switching Systems using Multi-Output Gaussian Processes

Mohammad Aali and Jun Liu

Abstract—Control barrier functions (CBFs) have recently been introduced as a systematic tool to ensure safety by establishing set invariance. When combined with a control Lyapunov function (CLF), they form a safety-critical control mechanism. However, the effectiveness of CBFs and CLFs is closely tied to the system model. In practice, model uncertainty can jeopardize safety and stability guarantees and may lead to undesirable performance. In this paper, we develop a safe learning-based control strategy for switching systems in the face of uncertainty. We focus on the case that a nominal model is available for a true underlying switching system. This uncertainty results in piecewise residuals for each switching surface, impacting the CLF and CBF constraints. We introduce a batch multi-output Gaussian process (MOGP) framework to approximate these piecewise residuals, thereby mitigating the adverse effects of uncertainty. A particular structure of the covariance function enables us to convert the MOGP-based chance constraints CLF and CBF into second-order cone constraints, which leads to a convex optimization. We analyze the feasibility of the resulting optimization and provide the necessary and sufficient conditions for feasibility. The effectiveness of the proposed strategy is validated through a simulation of a switching adaptive cruise control system.

I. INTRODUCTION

Many real-world control problems are modeled as hybrid dynamical systems, which involve a coupling between continuous dynamics and discrete events. Research in the stability analysis of such systems has grown significantly in recent decades [7], [10], [18]. More recently, Gaussian processes (GPs) are used to approximate hybrid residual dynamics model which can be integrated into a stabilizing model predictive control design ensuring probabilistic constraint satisfaction [11]. However, in the modern control applications, the focus has expanded from mere stability assurance to encompass safety verification. CBFs have recently provided a powerful theoretical tool for synthesizing controllers that ensure the safety of dynamical systems. They were initially applied to autonomous driving and bipedal walking [3]. Stability guarantees are then added via a unified control Lyapunov function-control barrier function (CLF-CBF) framework by solving a state-dependent quadratic programming (QP) problem at each time step to compute the control input [22]. In the context of hybrid systems, barrier functions were initially used as safety certificates in [21]. After the introduction of CBFs, CLF-CBF design has

been extensively used in safety-critical methods for hybrid systems [3], [19], [20]. However, there are very few studies on learning-based safety verification of switching systems via CBFs in the presence of uncertainty.

Learning-based CBFs for uncertain systems can be divided into two approaches. The first approach learns the unknown dynamics and uses the approximated model to derive safety certificates. In [24], a GP is employed to learn the unmodeled dynamics for the safe navigation of quadrotors with CBFs. A GP-based systematic controller synthesis method is introduced in [15], offering an inherently safe control strategy. In [12], the authors employ a Bayesian approach to learn model uncertainty while ensuring safety during the learning process. Learning the complete system dynamics with GPs is computationally demanding. Therefore, a common approach is to learn each component of the vector fields individually, disregarding their correlations. Furthermore, existing studies typically neglect uncertainty in the actuation term to simplify the learning-based model.

These limitations lead us to the second approach, which approximates the projection of the residual onto the CLF and CBF. In [9], an episodic learning method using neural networks is employed to approximate the impact of unmodeled dynamics of hybrid bi-pedal robot on system safety, eliminating the need for exhaustive data collection. However, methods employing neural networks must achieve precise approximations of the dynamics to formally ensure safety. In general, this requirement is challenging to meet, but GPs can provide uncertainty quantification with analytical confidence bounds. In [8], a GP-based min-norm controller stabilizing an unknown control affine system while guaranteeing safety using CLF-CBF method is introduced. The effect of the uncertainty in the stability and safety constraints is approximated by a GP model. This technique results in the approximation of a real-valued function instead of a vector field, leading to a significant reduction in the model complexity for high-dimensional systems. In [1], the effect of uncertainty on high-order CBFs is quantified using GPs.

In this paper, we develop a batch MOGP model to learn the piecewise residuals in each switching surface. We use the multi-output design to learn the piecewise residuals corresponding to CLF and CBF simultaneously, and the batched design facilitates for efficient computation of the model's hyperparameters. The main contributions of this paper relative to prior works are presented as follows:

- We show that the switching system imposes piecewise residuals on CLF and CBF constraints. We develop a batch MOGP model to efficiently approximate residuals

This work was supported in part by Nutrien Ltd., NSERC of Canada, and the Canada Research Chairs program.

Mohammad Aali is with the Department of Applied Mathematics, University of Waterloo, Canada, mohammad.aali@uwaterloo.ca

Jun Liu is with the Department of Applied Mathematics, University of Waterloo, Canada, j.liu@uwaterloo.ca

in each switching surface.

- We demonstrate that by selecting a particular form of the covariance function in the batch MOGP structure, the resulting min-norm controller with an uncertainty-aware chance constraint can be converted into a second-order cone program (SOCP).
- We analyze the feasibility of the resulting constrained optimization problem and determine the necessary and sufficient conditions for feasibility.

A conference version of this paper will appear in [2]. This extended version includes additional details omitted in the conference paper.

II. PROBLEM SETUP AND STATEMENT

Consider the control-affine switching system of the form

$$\begin{aligned} \dot{\mathbf{x}} &= \sum_{r=1}^R \delta_r (f_r(\mathbf{x}) + g_r(\mathbf{x})\mathbf{u}), \\ \delta_r &= \begin{cases} 1 & \text{if } \mathbf{x} \in \mathcal{R}_r, \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \quad (1)$$

where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ is the state, and $\mathbf{u} \in \mathbb{R}^m$ is the input, and \mathcal{X} is a compact and convex set. The function $\delta_r : \mathcal{X} \rightarrow \{0, 1\}$, $r \in \mathcal{I}$ is a state-dependent switching signal with the finite index set $\mathcal{I} = \{1, \dots, R\}$ which indicates the index of the active switching surface $\mathcal{R}_r \subset \mathcal{X}$. The vector fields $f_r : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g_r : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$, $r \in \mathcal{I}$, are the unknown dynamics, which are assumed to be locally Lipschitz in their arguments. Let $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be a locally Lipschitz continuous state feedback control law. We assume that the impulse effects are absent, i.e., the reset map is the identity. Furthermore, we assume that the closed-loop system

$$\dot{\mathbf{x}} = F_{cl}(\mathbf{x}) \triangleq \sum_{r=1}^R \delta_r (f_r(\mathbf{x}) + g_r(\mathbf{x})\pi(\mathbf{x})), \quad (2)$$

satisfies Caratheodory condition [14]. Then, for any initial state $\mathbf{x}(t_0) = \mathbf{x}_0$, system (2) admits a solution $\mathbf{x}(t)$ in the sense of Caratheodory defined on a maximal interval of existence $I(\mathbf{x}_0) = [t_0, I_{max})$. In this paper, we consider complete solutions, thus $I_{max} = \infty$.

Assumption 1: We assume that the state space is partitioned into R non-overlapping regions labeled as $\mathcal{R}_r \subset \mathcal{X}$, such that they cover the whole state space, i.e., $\cup_{r=1}^R \mathcal{R}_r = \mathcal{X}$, and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for all $i, j \in \{1, \dots, R\}$, $i \neq j$.

Assumption 1 requires system (1) to be well-posed, i.e., for all $\mathbf{x} \in \mathcal{X}$ there exists only one active index $r \in \mathcal{I}$ satisfying the membership condition.

A. Control Lyapunov Functions for Switching Systems

Definition 1 (Class \mathcal{K} function [16]): We say that a continuous function $\alpha : [0, a) \rightarrow [0, \infty)$, $a > 0$, belongs to class \mathcal{K} , if it is strictly increasing and $\alpha(0) = 0$.

Definition 2 (Control Lyapunov functions): A continuously differentiable function $V : \mathcal{X} \rightarrow \mathbb{R}$ is

a CLF for system (1) if there exist positive constants $\lambda, c_1, c_2 > 0$ such that for all $\mathbf{x} \in \mathcal{X}$ and $r \in \mathcal{I}$,

$$\begin{aligned} c_1 \|\mathbf{x}\|^2 &\leq V(\mathbf{x}) \leq c_2 \|\mathbf{x}\|^2, \\ \sum_{r=1}^R \delta_r (L_{f_r} V(\mathbf{x}) + L_{g_r} V(\mathbf{x})\mathbf{u}) &\leq -\lambda V(\mathbf{x}), \end{aligned} \quad (3)$$

where $L_{f_r} V(\mathbf{x})$ and $L_{g_r} V(\mathbf{x})$ are the Lie derivatives of $V(\mathbf{x})$ with respect to the corresponding vector fields.

Therefore, if the subsystems in (1) share a CLF, the rate of decrease of V along the vector fields, given by (3), is not affected by the switching, thus exponential stability is uniform with respect to δ_r . Also, if there exists a compact subset $D \subseteq \mathcal{X}$ including the origin such that (3) holds for some $\lambda > 0$, and there exists a sublevel set $\Omega = \{\mathbf{x} \in D \mid V(\mathbf{x}) \leq c\}$, where $c > 0$, then the origin is locally exponentially stabilizable from Ω .

B. Control Barrier Functions

CBF method defines safety based on the notion of set invariance, where a subset of the state space is specified as the safe set. This set is characterized by the zero-superlevel set of a continuously differentiable function $h : \mathbb{R}^n \rightarrow \mathbb{R}$ as

$$\begin{aligned} \mathcal{C} &= \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0\}, \\ \partial \mathcal{C} &= \{\mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = 0\}. \end{aligned} \quad (4)$$

CBFs provide a constructive tool for achieving forward invariance of set \mathcal{C} . We define a CBF as follows:

Definition 3 (Control barrier function [3]): Given the set \mathcal{C} as defined in (4), the continuously differentiable function $h(\mathbf{x})$ is called a CBF on a domain \mathcal{D} with $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$, if there exists a class \mathcal{K} function α such that the following holds for all $\mathbf{x} \in \mathcal{D}$ and $r \in \mathcal{I}$

$$\sum_{r=1}^R \delta_r (L_{f_r} h(\mathbf{x}) + L_{g_r} h(\mathbf{x})\mathbf{u}) \geq -\alpha(h(\mathbf{x})). \quad (5)$$

We can derive the following Corollary based on Nagumo's theorem [5] for the forward invariance of the safe set \mathcal{C} :

Corollary 1: Given CBF $h : \mathbb{R}^n \rightarrow \mathbb{R}$ with the associated set \mathcal{C} in (3), if $\nabla h(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in \partial \mathcal{C}$, any Lipschitz continuous controller $\mathbf{u}(\mathbf{x})$ satisfying (5), guarantees that the set \mathcal{C} is forward invariant for the system (1) and thus safe.

Inequalities (3) and (5) impose an affine condition on the control values which can be used to ensure stability and safety. Consider the CLF constraint (3), primarily designed to fulfill control objectives. Our goal is to apply the control law that satisfies (3) to the system only if it complies with (5). In practice, this is accomplished by solving the following QP optimization:

$$\mathbf{u}_s = \arg \min_{(\mathbf{u}, d) \in \mathbb{R}^{m+1}} \|\mathbf{u}\|_2^2 + \rho d^2 \quad (6a)$$

$$\text{s.t.} \quad \sum_{r=1}^R \delta_r (L_{f_r} V(\mathbf{x}) + L_{g_r} V(\mathbf{x})\mathbf{u}) + \lambda V(\mathbf{x}) \leq d, \quad (6b)$$

$$\sum_{r=1}^R \delta_r (L_{f_r} h(\mathbf{x}) + L_{g_r} h(\mathbf{x})\mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0, \quad (6c)$$

where ρ is a positive coefficient and $d \in \mathbb{R}$ is a slack variable. The resulting minimally invasive point-wise controller prioritizes safety over control objectives by satisfying (5) as a hard constraint.

C. Impact of uncertainty in the switching systems

In this section, we focus on reformulating CLF and CBF constraints such that they take the model uncertainty into account. We refer to the system (1) as the true system, which is unknown and modeled by a single nominal model in all regions as

$$\dot{\mathbf{x}} = \hat{f}(\mathbf{x}) + \hat{g}(\mathbf{x})\mathbf{u}, \quad (7)$$

where $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\hat{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are locally Lipschitz in \mathbf{x} . We design a locally exponentially stabilizing CLF and CBF based on the nominal system and assume they are also valid for the true system.

Due to the inherent model mismatch between the true system and the nominal model, the time derivatives of $V(\mathbf{x})$ and $h(\mathbf{x})$ based on the nominal model differ from their true values. The resulting error can be represented as

$$\begin{aligned} d^V(\mathbf{x}, \mathbf{u}) &= \dot{V}(\mathbf{x}, \mathbf{u}) - \hat{V}(\mathbf{x}, \mathbf{u}) = \sum_{r=1}^R \delta_r d_r^V(\mathbf{x}, \mathbf{u}), \\ d^h(\mathbf{x}, \mathbf{u}) &= \dot{h}(\mathbf{x}, \mathbf{u}) - \hat{h}(\mathbf{x}, \mathbf{u}) = \sum_{r=1}^R \delta_r d_r^h(\mathbf{x}, \mathbf{u}), \end{aligned}$$

where for $r \in \mathcal{I}$, we have

$$\begin{aligned} d_r^V &= L_{f_r} V(\mathbf{x}) - L_{\hat{f}} V(\mathbf{x}) + (L_{g_r} V(\mathbf{x}) - L_{\hat{g}} V(\mathbf{x})) \mathbf{u}, \\ d_r^h &= L_{f_r} h(\mathbf{x}) - L_{\hat{f}} h(\mathbf{x}) + (L_{g_r} h(\mathbf{x}) - L_{\hat{g}} h(\mathbf{x})) \mathbf{u}, \end{aligned} \quad (8)$$

where the dependence on \mathbf{x} and \mathbf{u} is removed for simplicity. Consequently, these discrepancies propagate into the CLF (6b) and CBF (6c) constraints, potentially jeopardizing the stability and safety of the system. Using (8), the constraints (6b) and (6c) become

$$L_{\hat{f}} V(\mathbf{x}) + L_{\hat{g}} V(\mathbf{x}) \mathbf{u} + \sum_{r=1}^R \delta_r d_r^V + \lambda V(\mathbf{x}) \leq d, \quad (9)$$

$$L_{\hat{f}} h(\mathbf{x}) + L_{\hat{g}} h(\mathbf{x}) \mathbf{u} + \sum_{r=1}^R \delta_r d_r^h + \alpha(h(\mathbf{x})) \geq 0. \quad (10)$$

In this paper, our primary focus is characterizing the adverse effects of uncertainty in the proposed safety-critical control method. We then adopt a supervised learning approach to make our design robust to piecewise residuals of the form (8). Thus, a dataset that contains information about piecewise residuals in all regions is needed.

Assumption 2: We have access to the set of samples $\{(\mathbf{x}(t), \mathbf{u}(t))\}, t \in [t_0, t_f]$ collected with sampling time Δt from trajectories of the system. We assume this set is rich and contains states from all regions $\mathcal{R}_r, r \in \mathcal{I}$.

Given these data samples, we can approximately measure d^V and d^h by collecting trajectories from the true system and

using the finite difference method to obtain

$$\begin{aligned} \omega_j^V &= (V(\mathbf{x}(t_j + \Delta t)) - V(\mathbf{x}(t_j))) / \Delta t - \hat{V}(\mathbf{x}_j, \mathbf{u}_j), \\ \omega_j^h &= (h(\mathbf{x}(t_j + \Delta t)) - h(\mathbf{x}(t_j))) / \Delta t - \hat{h}(\mathbf{x}_j, \mathbf{u}_j), \end{aligned} \quad (11)$$

where $t_j = t_0 + (j-1)\Delta t$, $\mathbf{x}_j = (\mathbf{x}(t_j + \Delta t) + \mathbf{x}(t_j)) / 2$ is the mean of the state, and $\mathbf{u}_j = \mathbf{u}(t_j)$ is the control input for $t_j \in [t_j, t_j + \Delta t], j \in \{1, \dots, N\}$.

A dataset can be generated by collecting trajectories of the true system. We denote it by $\mathcal{D} = \{((\mathbf{x}_j, \mathbf{u}_j), \omega_j^V, \omega_j^h)\}_{j=1}^N$, where $(\mathbf{x}_j, \mathbf{u}_j) \in \mathcal{X} \times \mathbb{R}^m$ is the input data and outputs ω_j^V and ω_j^h are obtained from (11). We will use this dataset to learn piecewise residuals.

III. PROPOSED BATCH MOGP-BASED DESIGN FOR PIECEWISE RESIDUALS

A. Single-output GP for real-valued functions

Typical single-output GP (SOGP) provides a framework to approximate nonlinear functions. We represent an SOGP approximation of $v : \mathcal{X} \rightarrow \mathbb{R}$ as $v(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, which is fully specified by its mean function $m : \mathcal{X} \rightarrow \mathbb{R}$ and covariance (kernel) function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Without loss of generality, we consider zero prior mean. Our prior knowledge of the problem can shape the form of $k(\mathbf{x}, \mathbf{x}')$, considering that the covariance function must satisfy positive definiteness [25]. Given noisy measurements $w_j = v(\mathbf{x}_j) + \varepsilon_j, j \in \{1, \dots, N\}$, which are corrupted by Gaussian noise $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$, a GP model can infer a posterior mean and variance for a test point \mathbf{x}_t conditioned on the measurements

$$\begin{aligned} \mu(\mathbf{x}_t) &= \mathbf{w}^T (K + \sigma_n^2 I)^{-1} \bar{K}^T, \\ \sigma(\mathbf{x}_t)^2 &= k(\mathbf{x}_t, \mathbf{x}_t) - \bar{K} (K + \sigma_n^2 I)^{-1} \bar{K}^T, \end{aligned} \quad (12)$$

where $\mathbf{w} \in \mathbb{R}^N$ is the vector of measurements w_j , $K \in \mathbb{R}^{N \times N}$ is the Gram matrix with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and $\bar{K} = [k(\mathbf{x}_t, \mathbf{x}_1), \dots, k(\mathbf{x}_t, \mathbf{x}_N)]^T \in \mathbb{R}^N$.

B. Batch multi-output GP for piecewise residuals

Going back to the original problem, our objective is to build a Bayesian approximation uncorrelated target functions d^V and d^h . We need to approximate each individual d_r^V and d_r^h , for $r \in \mathcal{I}$. Since the input data $(\mathbf{x}_j, \mathbf{u}_j)$ gathered from the same trajectory, we leverage from batch multi-output structure to learn d_r^V and d_r^h simultaneously. As d_r^V and d_r^h are uncorrelated, we consider an independent MOGP framework. This design facilitates more computationally efficient GP training. We initially partition the dataset \mathcal{D} , as described in Section II into R datasets of the form

$$\mathcal{D}_r = \{((\mathbf{x}_j, \mathbf{u}_j), \omega_j^V, \omega_j^h) | ((\mathbf{x}_j, \mathbf{u}_j), \omega_j^V, \omega_j^h) \in \mathcal{D}, \mathbf{x}_j \in \mathcal{R}_r\}$$

for $r \in \mathcal{I}$, and denote the cardinality of each set by N_r . In a batch GP, each dataset or batch is associated with its own GP, and these GPs can capture the specific behavior and dependencies within each batch. Now, we employ MOGP to approximate R individual GPs for d_r^V and d_r^h

simultaneously within a unified framework. Let $\mathbf{d}_r(\mathbf{x}, \mathbf{u}) = [d_r^V(\mathbf{x}, \mathbf{u}) \ d_r^h(\mathbf{x}, \mathbf{u})]^T \in \mathbb{R}^2$ be the vector-valued function which follows an MOGP as

$$\mathbf{d}_r(\mathbf{x}, \mathbf{u}) \sim \mathcal{GP}^r \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_r^V & 0 \\ 0 & k_r^h \end{bmatrix} \right). \quad (13)$$

where the dependence to (\mathbf{x}, \mathbf{u}) for the covariance functions is removed for simpler notation. As a result, training a MOGP model (13) leading to separate posterior distributions for target functions d_r^V and d_r^h :

$$\begin{aligned} d_r^V(\mathbf{x}, \mathbf{u}) &\sim \mathcal{N}(m_r^V(\mathcal{D}_r, \mathbf{x}, \mathbf{u}), \sigma_r^{V^2}(\mathcal{D}_r, \mathbf{x}, \mathbf{u})), \\ d_r^h(\mathbf{x}, \mathbf{u}) &\sim \mathcal{N}(m_r^h(\mathcal{D}_r, \mathbf{x}, \mathbf{u}), \sigma_r^{h^2}(\mathcal{D}_r, \mathbf{x}, \mathbf{u})). \end{aligned} \quad (14)$$

From (8), we can verify that the residuals are control-affine and can be characterized by

$$d_r^V = [L_{f_r}V - L_{\hat{f}}V \quad L_{g_r}V - L_{\hat{g}}V] \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} = \boldsymbol{\varphi}_r^V \mathbf{y}, \quad (15)$$

$$d_r^h = [L_{f_r}h - L_{\hat{f}}h \quad L_{g_r}h - L_{\hat{g}}h] \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} = \boldsymbol{\varphi}_r^h \mathbf{y}. \quad (16)$$

We denote the concatenation of one and \mathbf{u} by $\mathbf{y} = [1, \mathbf{u}^T]^T \in \mathbb{R}^{m+1}$ and $\boldsymbol{\varphi}_r^V, \boldsymbol{\varphi}_r^h \in \mathbb{R}^{1 \times m+1}$. This prior knowledge of the problem can be embedded into the structure of the kernel function.

Proposition 1: Let $\mathbf{x} \in \mathcal{R}_r$, $\mathbf{y} \in \mathbb{R}^{m+1}$, and define the input domain $\tilde{\mathcal{X}}_r = \mathcal{R}_r \times \mathbb{R}^{m+1}$ for a given $r \in \mathcal{I}$. Then, consider the real-valued function $k_r : \tilde{\mathcal{X}}_r \times \tilde{\mathcal{X}}_r \rightarrow \mathbb{R}$, defined by

$$k_r \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} \right) = \mathbf{y}^T \Lambda_r(\mathbf{x}, \mathbf{x}') \mathbf{y}', \quad (17)$$

where $\Lambda_r(\mathbf{x}, \mathbf{x}') = \text{diag}([k_r^1(\mathbf{x}, \mathbf{x}'), \dots, k_r^{m+1}(\mathbf{x}, \mathbf{x}')])$ and $k_r^i : \mathcal{R}_r \times \mathcal{R}_r \rightarrow \mathbb{R}, i \in \{1, \dots, m+1\}$, we denote the individual components k_r^i 's as base kernels. If base kernels are all positive-definite, then k_r is a positive-definite kernel.

Proof: Since $k_r^i(\mathbf{x}, \mathbf{x}')$'s are positive definite kernels, then by the definition, there is a feature map $\varphi(\mathbf{x})$ such that $k_r^i(\mathbf{x}, \mathbf{x}') = \varphi_i^T(\mathbf{x})\varphi_i(\mathbf{x}')$ for $i \in \{1, \dots, m+1\}$ [4]. Now, Let y^i and y'^i be the i^{th} element of the corresponding vectors. We have

$$\begin{aligned} k_r &= \mathbf{y}^T \text{diag}([\varphi_1^T(\mathbf{x})\varphi_1(\mathbf{x}'), \dots, \varphi_{m+1}^T(\mathbf{x})\varphi_{m+1}(\mathbf{x}')]) \mathbf{y}' \\ &= \sum_{i=1}^{m+1} y^i \varphi_i^T(\mathbf{x}) \varphi_i(\mathbf{x}') y'^i \\ &= [y^1 \varphi_1^T(\mathbf{x}) \quad \dots \quad y^{m+1} \varphi_{m+1}^T(\mathbf{x})] \begin{bmatrix} y'^1 \varphi_1(\mathbf{x}') \\ \vdots \\ y'^{m+1} \varphi_{m+1}(\mathbf{x}') \end{bmatrix} \\ &= \psi^T \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \right) \psi \left(\begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} \right), \end{aligned}$$

which again by the definition of kernels, proves that $k_r(\cdot, \cdot)$ is a positive-definite kernel. ■

An immediate consequence of Proposition 1 is that k_r is the reproducing kernel of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_{k_r}(\tilde{\mathcal{X}})$. Also, this structure allows us to choose k_r^i 's specifically for each region $r \in \mathcal{I}$.

We denote the input samples and domain by $(\mathbf{x}, \mathbf{y}) \in \tilde{\mathcal{X}}$ as defined above. Let $X_r \in \mathbb{R}^{n \times N_r}$ and $Y_r \in \mathbb{R}^{(m+1) \times N_r}$ be matrices whose columns are vectors \mathbf{x}_j and \mathbf{y}_j of the corresponding dataset \mathcal{D}_r and let outputs $\boldsymbol{\omega}_r \in \mathbb{R}^{N_r}$ be the stacked vector of the corresponding outputs to the states $\mathbf{x}_j \in \mathcal{R}_r$. Based on the collected batches, the MOGP model (13) equipped with the kernels of the form k_r in Proposition 1, gives the following expression for the posterior distribution of a test point $(\mathbf{x}_*, \mathbf{y}_*) \in \tilde{\mathcal{X}}_r$:

$$m_r^* = \boldsymbol{\omega}_r^T (k_r + \sigma_n^2 I)^{-1} \bar{K}_r^T \mathbf{y}_*, \quad (18)$$

$$\sigma_r^{*2} = \mathbf{y}_*^T (\Lambda_r(\mathbf{x}_*, \mathbf{x}_*) - \bar{K}_r (K_r + \sigma_n^2 I)^{-1} \bar{K}_r^T) \mathbf{y}_*, \quad (19)$$

where K_r is the Gram matrix of $k_r(\cdot, \cdot)$ for the input data pair (X_r, Y_r) , and $\bar{K}_r \in \mathbb{R}^{(m+1) \times N_r}$ is given by

$$\begin{aligned} \bar{K}_r &= [\bar{\mathbf{k}}_r^1 \quad \bar{\mathbf{k}}_r^2 \quad \dots \quad \bar{\mathbf{k}}_r^{N_r}] \circ Y_r, \\ \bar{\mathbf{k}}_r^i &= [k_r^1(\mathbf{x}_*, \mathbf{x}_i) \quad \dots \quad k_r^{m+1}(\mathbf{x}_*, \mathbf{x}_i)]^T, i \in \{1, \dots, N_r\}, \end{aligned}$$

where \circ denotes the element-wise multiplication (Hadamard product) of two matrices with identical dimensions. The posterior mean and variance of d_r^V and d_r^h are obtained by substituting the output vectors $\boldsymbol{\omega}_r^V$ and $\boldsymbol{\omega}_r^h$ and their corresponding kernel structure into (18) and (19).

This result highlights a key advantage of using the proposed kernel. The resulting expression for the posterior mean $m_r^* = \boldsymbol{\mu}_r(\mathbf{x}_*) \mathbf{y}_*$ and the variance $\sigma_r^{*2} = \mathbf{y}_*^T \Sigma_r(\mathbf{x}_*) \mathbf{y}_*$ are linear and quadratic in \mathbf{y}_* (and the control input), respectively. We will exploit this feature in the next section to establish a convex GP-based safety filter.

C. Confidence bounds for the estimation of the effect of uncertainty

Although GP is inherently a probabilistic model, a high probability error bound can be derived for the distance between the true value and the GP prediction. This requires an additional assumption on d_r^V and d_r^h . Let d_r be either d_r^V or d_r^h , and φ_r be either φ_r^V or φ_r^h from (15) and (16).

Assumption 3: We assume that each i^{th} element of φ_r is a member of $\mathcal{H}_{k_r^i}$ for $i \in \{1, \dots, m+1\}$ with bounded RKHS norm $\|\varphi_r^i\|_{k_r^i} \leq \eta_r$ for $i \in \{1, \dots, m+1\}$ within a given region $r \in \mathcal{I}$.

Lemma 1 ([23]): Let Assumption 3 hold. Then, with a probability of at least $1 - \delta$, the following holds

$$|m_r^* - d_r(\mathbf{x}, \mathbf{u})| \leq \beta_r \sigma_r^*, \quad (20)$$

on $\mathcal{R}_r \subset \mathcal{X}$, $\delta \in (0, 1)$ and

$$\beta_r = \sqrt{2\eta_r^2 + 300\kappa_{N_r+1} \log^3((N_r + 1)/\delta)},$$

where κ_{N_r+1} is the maximum mutual information that can be obtained after getting $N_r + 1$ data, and η_r is the upper bound of the corresponding RKHS norm, and m_r^* and σ_r^* are the posterior mean and standard deviation of a test point $(\mathbf{x}_*, \mathbf{y}_*) \in \tilde{\mathcal{X}}_r$.

Based on the probabilistic bounds on (20) and the CLF and CBF constraints (9) and (10), we can conclude that the

following hold with a probability of at least $1 - \delta$

$$\dot{V}(\mathbf{x}, \mathbf{u}) \leq \hat{V}(\mathbf{x}, \mathbf{u}) + \sum_{r=1}^R \delta_r (m_r^V + \beta_r \sigma_r^V), \quad (21)$$

$$\dot{h}(\mathbf{x}, \mathbf{u}) \geq \hat{h}(\mathbf{x}, \mathbf{u}) + \sum_{r=1}^R \delta_r (m_r^h - \beta_r \sigma_r^h), \quad (22)$$

where $m_r^V(\mathbf{x}, \mathbf{y})$ and $\sigma_r^V(\mathbf{x}, \mathbf{y})$ are the mean and standard deviation of a query point derived from (18), (19). Incorporating (21) and (22) into the optimization problem (6), we have

$$\mathbf{u}_s = \arg \min_{(\mathbf{u}, d) \in \mathbb{R}^{m+1}} \|\mathbf{u}\|_2^2 + \rho d^2 \quad (23a)$$

$$\text{s.t. } \hat{V}(\mathbf{x}, \mathbf{u}) + \sum_{r=1}^R \delta_r (m_r^V + \beta_r \sigma_r^V) + \lambda V(\mathbf{x}) \leq d, \quad (23b)$$

$$\hat{h}(\mathbf{x}, \mathbf{u}) + \sum_{r=1}^R \delta_r (m_r^h - \beta_r \sigma_r^h) + \alpha(h(\mathbf{x})) \geq 0, \quad (23c)$$

Note that the constraint in (23) is constructed regardless of the underlying true dynamics.

Theorem 1: The optimization problem (23) is convex and can be converted into the standard SOCP of the form (24), if m_r^V, m_r^h , and σ_r^V, σ_r^h satisfy (18) and (19), respectively.

$$\mathbf{u}_{\text{socp}} = \arg \min_z \mathbf{f}^T \mathbf{z} \quad \text{s.t. } \sum_{r=1}^R \delta_r \|M_r^i \mathbf{z} + \mathbf{n}_r^i\|_2 \leq \sum_{r=1}^R \delta_r (\mathbf{p}_r^{iT} \mathbf{z} + q_r^i), \quad (24)$$

for $i = 1, \dots, n_c$, where $\mathbf{z} \in \mathbb{R}^{n_z}$ and $\mathbf{f} \in \mathbb{R}^{n_z}$, and the matrix M_r^i and $\mathbf{n}_r^i, \mathbf{p}_r^i, q_r^i$ have appropriate dimensions for n_c constraints.

Proof: Let's denote the objective function of (23) by $J = \|\mathbf{u}\|_2^2 + \rho d^2$. Let $\mathbf{p} = [\mathbf{u}^T \ d]^T \in \mathbb{R}^{m+1}$. Then, we can rewrite J as the equivalent form of $J_1 = \mathbf{p}^T Q \mathbf{p}$, where Q is a positive definite diagonal matrix with matrix square root E . Then, we convert J_1 into $J_2 = \|E\mathbf{p}\|_2^2$. Since the Euclidean norm is always positive, we can consider the equivalent objective function $J_3 = \|E\mathbf{p}\|_2$ and set $\|E\mathbf{p}\|_2 \leq t$, where $t \in \mathbb{R}$ is an auxiliary variable. Then, we convert J_3 to a second-order cone constraint (25). Now, we need to solve a new minimization problem with a new augmented variable $\mathbf{z} = [\mathbf{u}^T \ d \ t]^T \in \mathbb{R}^{m+2}$ as

$$\begin{aligned} & \min [\mathbf{0}_{m+1}^T \ 1] \mathbf{z} \\ & \text{s.t. } \underbrace{\| [E \ \mathbf{0}_{m+1}] \mathbf{z} \|_2}_{:=M_r^1} \leq \underbrace{[\mathbf{0}_{m+1}^T \ 1] \mathbf{z}}_{:=\mathbf{p}_r^{1T}}, \end{aligned} \quad (25)$$

where $\mathbf{0}_{m+1} \in \mathbb{R}^{m+1}$ is a vector of zeros and $\mathbf{f} = [\mathbf{0}_{m+1}^T \ 1]^T$.

Next, we need to show that constraints (23b) and (23c) are SOC constraints. We need to consider the constraints for a given region index $r \in \mathcal{I}$ as at each time step one region is active according to the defined switching signal δ_r in Section

I. As the proof is the same for both constraints, we consider the CBF constraint (23c). Note that based on (18), we have $m_r^h(\mathbf{x}, \mathbf{y}) = \boldsymbol{\mu}_r^h(\mathbf{x})\mathbf{y}$, which can be written as

$$m_r^h(\mathbf{x}, \mathbf{y}) = \mu_r^{h1} + \boldsymbol{\mu}_r^{hm} \mathbf{u},$$

where the notation μ_r^{h1} refers the first element and $\boldsymbol{\mu}_r^{hm}$ refers the last m elements of the row vector $\boldsymbol{\mu}_r^h \in \mathbb{R}^{1 \times (m+1)}$. Also, from (16), we know that φ_r^h is control affine, i.e. $\varphi_r^h = \varphi_r^{hf} + \varphi_r^{hg} \mathbf{u}$, where $\varphi_r^{hf} \in \mathbb{R}$ and the row vectors $\varphi_r^{hg} \in \mathbb{R}^{1 \times m}$. Hence, the right hand side of the inequality in (23c) is affine in \mathbf{u} as desired.

Based on (19), we have that $\sigma_r^{h2}(\mathbf{x}, \mathbf{y}) = \mathbf{y}^T \Sigma_r^h(\mathbf{x}) \mathbf{y}$, where $\Sigma_r^h \in \mathbb{R}^{(m+1) \times (m+1)}$. Since k_r is a valid kernel and $\sigma_n > 0$, Σ_r^h is positive definite. Then, we have

$$\sigma_r^h(\mathbf{x}, \mathbf{y}) = \sqrt{\mathbf{y}^T \Sigma_r^h \mathbf{y}} = \sqrt{\mathbf{y}^T L_r^T L_r \mathbf{y}} = \|L_r \mathbf{y}\|_2,$$

where $L_r \in \mathbb{R}^{(m+1) \times (m+1)}$ is the matrix square root of Σ_r^h . By the definition $\mathbf{y} = [\mathbf{1} \ \mathbf{u}^T]^T$, we have $\sigma_r^h(\mathbf{x}, \mathbf{y}) = \|\mathbf{l}_r^1 + L_r^m \mathbf{u}\|_2$, where the notation \mathbf{l}_r^1 , and L_r^m refers to the first column and the last m columns of the matrix L_r , respectively. Now, we can rewrite the safety certificate (23c) as a SOC constraint of the form

$$\|A_r^h(\mathbf{x})\mathbf{u} + \mathbf{b}_r^h(\mathbf{x})\|_2 \leq \mathbf{c}_r^h(\mathbf{x})\mathbf{u} + d_r^h(\mathbf{x}), \quad (26)$$

where

$$\begin{aligned} A_r^h(\mathbf{x}) &= \beta_r L_r^m \in \mathbb{R}^{(m+1) \times (m)}, \\ \mathbf{b}_r^h(\mathbf{x}) &= \beta_r \mathbf{l}_r^1 \in \mathbb{R}^{m+1}, \\ \mathbf{c}_r^h(\mathbf{x}) &= \varphi_r^{hg} + \boldsymbol{\mu}_r^{hm} \in \mathbb{R}^{1 \times m} \\ d_r^h(\mathbf{x}) &= \varphi_r^{hf} + \mu_r^{h1} \in \mathbb{R}. \end{aligned} \quad (27)$$

It can be expressed in terms of the new variable \mathbf{z} as

$$\underbrace{\|A_r^h(\mathbf{x}) \ \mathbf{0}_{m+1}\|_2}_{:=M_r^2} \mathbf{z} + \underbrace{\|\mathbf{b}_r^h(\mathbf{x})\|_2}_{:=n_r^2} \leq \underbrace{\|\mathbf{c}_r^h(\mathbf{x}) \ 0\|_2}_{:=\mathbf{p}_r^{2T} \mathbf{z}} + \underbrace{d_r^h(\mathbf{x})}_{:=q_r^2}.$$

The same proof holds for the CLF constraint (23b). Adding the resulting constraints to (25) leads to (24), which concludes the proof. \blacksquare

D. Feasibility analysis

In this section, we will analyze the point-wise feasibility of the SOCP (24). As the CLF is considered as a soft constraint, only the CBF constraint restricts the feasibility. Intuitively, if the batch MOGP prediction is less accurate, the resulting standard deviation $\sigma_r^h(\mathbf{x}, \mathbf{y})$ will be large, leading to a more conservative approach. In particular, it restricts the space on which a safe control input can be selected, which may make the SOCP infeasible. In the following, we theoretically analyze the conditions for feasibility.

Theorem 2: Given a state $\mathbf{x} \in \mathcal{R}_r$, the SOCP (24) is feasible if and only if there exists a control input $\mathbf{u} \in \mathbb{R}^m$ that satisfies the following conditions

$$[d_r^h(\mathbf{x}) \ \mathbf{c}_r^h(\mathbf{x})] \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} \geq 0, \quad (28)$$

$$[1 \ \mathbf{u}^T] S_r(\mathbf{x}) \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} \leq 0, \quad (29)$$

where $S_r \in \mathbb{R}^{(m+1) \times (m+1)}$ is of the form

$$\begin{aligned} S_r(\mathbf{x}) &= \begin{bmatrix} S_r^1(\mathbf{x}) & S_r^2(\mathbf{x}) \\ S_r^{2T}(\mathbf{x}) & S_r^3(\mathbf{x}) \end{bmatrix}, \\ S_r^1(\mathbf{x}) &= \mathbf{b}_r^h(\mathbf{x})\mathbf{b}_r^h(\mathbf{x}) - d_r^h(\mathbf{x})d_r^h(\mathbf{x}), \\ S_r^2(\mathbf{x}) &= \mathbf{b}_r^h(\mathbf{x})A_r^h(\mathbf{x}) - d_r^h(\mathbf{x})c_r^h(\mathbf{x}), \\ S_r^3(\mathbf{x}) &= A_r^h(\mathbf{x})^T A_r^h(\mathbf{x}) - c_r^h(\mathbf{x})^T c_r^h(\mathbf{x}), \end{aligned} \quad (30)$$

with $A_r^h(\mathbf{x})$, $\mathbf{b}_r^h(\mathbf{x})$, $c_r^h(\mathbf{x})$, and $d_r^h(\mathbf{x})$ are defined in (27).

Proof: In order to analyze the feasibility of the SOCP (24), we need to verify that the safety SOC constraint (26) is feasible. Since the left hand side is always non-negative, the right hand side must be also non-negative, which leads to the first condition.

Now, since both sides of (26) are non-negative, we can take squares of both sides and write the resulting expression in quadratic form. and collect all the terms on the left. Factorizing the similar terms gives the second condition. ■ Based on this result, we obtain a necessary condition for point-wise feasibility as stated in the following:

Corollary 2: Given $\mathbf{x} \in \mathcal{R}_r$, if the SOCP (24) is feasible, then the following condition must be satisfied

$$1 - \frac{1}{\beta_r^2} \phi_r \Sigma_r^{h-1} \phi_r^T \leq 0, \quad (31)$$

where $\phi_r = [\varphi_r^{hf} + \mu_r^{h1} \quad c_r^h(\mathbf{x})] \in \mathbb{R}^{1 \times (m+1)}$.

Proof: We prove this by contradiction. Let's assume that there exists a solution $\mathbf{u} \in \mathbb{R}^m$ of the SOCP (24), which satisfies $1 - \frac{1}{\beta_r^2} \phi_r \Sigma_r^{h-1} \phi_r^T > 0$. Let's define the matrix M_r partitioned as

$$M_r = \begin{bmatrix} 1 & \phi_r \\ \phi_r^T & \beta_r^2 \Sigma_r^h(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{(m+2) \times (m+2)}.$$

The Schur complements of M_r is obtained by

$$\begin{cases} M_r/1 = \beta_r^2 \Sigma_r^h(\mathbf{x}) - \phi_r^T \phi_r, \\ M_r/\beta_r^2 \Sigma_r^h(\mathbf{x}) = 1 - \frac{1}{\beta_r^2} \phi_r \Sigma_r^{h-1} \phi_r^T. \end{cases}$$

We can easily verify that $\beta_r^2 \Sigma_r^h = [\beta_r \mathbf{l}_r^1 \quad A_r^h(\mathbf{x})]^T [\beta_r \mathbf{l}_r^1 \quad A_r^h(\mathbf{x})]$ and $M_r/1 = S_r(\mathbf{x})$. We know that $\Sigma_r^h(\mathbf{x})$ is symmetric positive-definite (SPD). So, M_r is also symmetric and we can characterize its definiteness by Schur complement theorem. Since its lower right block is positive definite and its corresponding Schur complement $M_r/\beta_r^2 \Sigma_r^h(\mathbf{x}) > 0$ by our assumption, based on [6], we can conclude that M_r is also positive definite.

Now, given that M_r is SPD, and its upper left block is always positive definite, we can conclude that $M_r/1 = S_r(\mathbf{x})$ must be positive definite, which is a contradiction to the condition (29). Thus, $1 - \frac{1}{\beta_r^2} \phi_r \Sigma_r^{h-1} \phi_r^T \leq 0$ holds. ■

Theorem 2 also provides the theoretical background to obtain a sufficient condition of feasibility.

Corollary 3: Given $\mathbf{x} \in \mathcal{R}_r$, the SOCP (24) is feasible at \mathbf{x} , if $S_r^3(\mathbf{x})$ is negative definite.

Proof: From (30), we know that $S_r^3(\mathbf{x})$ is a symmetric matrix, thus it has real eigenvalues. Let's denote its maximum eigenvalue and the corresponding eigenvector by λ_r^m , \mathbf{e}_r^m .

Then, by the assumption that $S_r^3(\mathbf{x})$ is negative definite, we can conclude that $\lambda_r^m < 0$ and $\mathbf{e}_r^{mT} S_r^3(\mathbf{x}) \mathbf{e}_r^m < 0$. Substituting (30) into the former, we have

$$\mathbf{e}_r^{mT} A_r^h(\mathbf{x}) A_r^h(\mathbf{x}) \mathbf{e}_r^m - (\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m)^T (\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m) < 0.$$

Since $A_r^h(\mathbf{x})^T A_r^h(\mathbf{x})$ is positive definite, $\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m \neq 0$ must hold. We take a control input in the direction of this eigenvector as $\mathbf{u}_e = \alpha \text{sgn}(\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m) \cdot \mathbf{e}_r^m$, $\alpha > 0$. Next, we need to check if the resulting control input can satisfy the conditions of Theorem 2. Substituting \mathbf{u}_e in (29), we have

$$S_r^1 + 2\alpha S_r^2((\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m) \cdot \mathbf{e}_r^m) + \alpha^2 \mathbf{e}_r^{mT} S_r^3(\mathbf{x}) \mathbf{e}_r^m.$$

By choosing large enough α , the above statement can be made negative since $\mathbf{e}_r^{mT} S_r^3(\mathbf{x}) \mathbf{e}_r^m < 0$. Also, substituting \mathbf{u}_e into (28), we have

$$\mathbf{c}_r^h(\mathbf{x}) \mathbf{u}_e + (\varphi_r^{hf} + \mu_r^{h1}) = \alpha |\mathbf{c}_r^h(\mathbf{x}) \mathbf{e}_r^m| + (\varphi_r^{hf} + \mu_r^{h1}).$$

Again, by choosing a large enough α , we can make the above expression positive, which concludes the proof. ■

E. Data collection and model fitting

We use an episodic data collection method. In the first episode, we run an initial roll-out using the CLF and CBF designed based on the nominal system. Using nominal system (7), we run QP controller (6) with the nominal values of the time derivative of $V(\mathbf{x})$ and $h(\mathbf{x})$. Then, we record the system's trajectory until it reaches an unsafe state. Next, we partition the resulting dataset into R datasets \mathcal{D}_r as described in Section III-B. Subsequently, we fit MOGPs configured with the kernel functions k_r to batches \mathcal{D}_r , for $r = 1, \dots, R$. To use (18) and (19) for prediction, we need to infer the hyperparameters θ_r for each individual kernel $k_r^i, i \in \{1, \dots, m+1\}$ in the structure of k_r . Each set of parameters is obtained by minimizing the negative log marginal likelihood [17] as

$$\theta_r^{\text{opt}} = \arg \min_{\theta_r} -\log p(\omega_r | \bar{X}_r, \theta_r), \quad (32)$$

where for the $J_{\theta_r} = -\log p(\omega_r | \bar{X}_r, \theta_r)$, we have

$$\begin{aligned} J_{\theta_r} &= \frac{1}{2} \omega_r^T [K_r(\bar{X}_r, \bar{X}_r) + \sigma_n^2 I]^{-1} \omega_r \\ &\quad + \frac{1}{2} \log |K_r(\bar{X}_r, \bar{X}_r) + \sigma_n^2 I| + \frac{N_r}{2} \log 2\pi, \end{aligned}$$

where $\bar{X}_r = (X_r, Y_r)$ such that $((X_r, Y_r), \omega_r) \in \mathcal{D}_r$. ω_r is either the vector of outputs ω_j^V or ω_j^h for approximating d_r^V or d_r^h , respectively. Then we use the uncertainty-aware SOCP optimization (23) with a high probability bound $(1 - \delta = 0.95)$ and run the system with the proposed method until the system reaches an unsafe state or the problem becomes infeasible. The data collected during each episode will be added to the dataset and the hyperparameters θ_r will be optimized using (32) with the new dataset. We repeat this process until the simulation completed without encountering any unsafe behavior.

IV. SIMULATION RESULTS

In this section, we highlight the effectiveness of the proposed method by applying it to a switching adaptive cruise control (ACC) system that is moving on varying road conditions. For comparison, we also implemented a baseline GP-CBF-CLF-SOCP [8] that fits a single GP to piecewise residuals and a nominal QP controller. The GP models were trained using GPyTorch [13].

Consider a switching ACC system modeled by

$$\dot{\mathbf{x}} = \sum_{r=1}^R \delta_r (f_r(\mathbf{x}) + g_r(\mathbf{x})) u, \quad (33)$$

$$f_r(\mathbf{x}) = \begin{bmatrix} x_2 \\ -\frac{F_r(x_2)}{m} \end{bmatrix}, g_r(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{c_r}{m} \end{bmatrix},$$

where the system state $\mathbf{x} = [x_1 \ x_2]^T \in \mathcal{X}$ containing the position of the ego car x_1 , and its forward velocity x_2 in the state space $\mathcal{X} = [0, 700] \times [0, 30]$. The control input is the wheel force $u \in \mathbb{R}$. The mass of the ego car is denoted by m and the rolling resistance is modeled by $F_r(x_2) = f_r^0 + f_r^1 x_2 + f_r^2 x_2^2$.

The switching behavior of the system arises from the changing road conditions which impact the rolling resistance $F_r(x_2)$ and traction control of the vehicle c_r . As the vehicle encounters different road surfaces $F_r(x_2)$ and c_r will change, leading to variations in the system dynamics. We consider hyper-rectangular partitions of \mathcal{X} with bounds $\mathcal{R}_1 = ([0, 15] \cup (25, 700]) \times [0, 30]$, and $\mathcal{R}_2 = [15, 25] \times [0, 30]$ that represent two different road conditions. The state-dependent switching signal δ_r in (1) determines the active region index at each time step. We only have access to the nominal model and the true system is unknown in both operating regions. The parameters for each model are shown in Table I.

TABLE I
PARAMETERS FOR THE NOMINAL AND TRUE SYSTEM IN $\mathcal{R}_1, \mathcal{R}_2$

	$m(kg)$	f_1^0	f_1^1	f_1^2	c_r
True \mathcal{R}_1	3300	0.2	10	0.5	1
True \mathcal{R}_2	3300	1	50	4.5	0.5
Nominal	1050	0.1	15	2.25	1

The distance between the ego car and the front car, denoted by z , is governed by the equation $\dot{z} = v_0 - x_2$, where $v_0 = 10 \text{ m/s}$ represents the constant velocity of the front car. The control objective is to achieve a target speed of v_d while ensuring a safe distance from the front vehicle. To stabilize the velocity at the desired value, a Lyapunov function $V(\mathbf{x}) = (x_2 - v_d)^2$ is considered. Additionally, a CBF $h(\mathbf{x}) = z - T_h x_2$, with $T_h = 1.6$, is utilized to maintain a safe distance proportional to the current velocity.

We selected infinitely differentiable squared exponential kernel [25], for $k_i, i = 1, 2$ base kernels in the configuration of k_r . We used the process in Section III-E to collect the dataset and optimize the batch MOGP model hyperparameters, which collected a total number of 390 data samples in the initial roll-out and a total of 999 data samples in all episodes. The simulation is started from the initial state

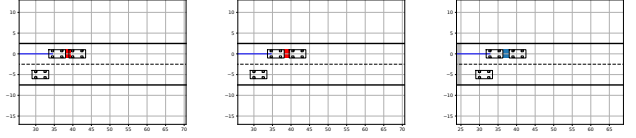


Fig. 1. Snapshots of the simulation to show the unsafe (red color) and safe (green color) distance between cars in different implementations. The nominal QP controller (left) and single GP-SOCP controller (middle) violate the safety distance. MOGP-based controller (right) ensures safety. See <https://youtu.be/8nmcIUSGJE> for the simulation video.

$\mathbf{x}_0 = [0 \ 14]^T$ and $z_0 = 140 \text{ m}$. The ego car starts in the region \mathcal{R}_1 from a safe distance behind the front car. First, the controller increases the velocity to converge to the desired speed resulting in a decrease in z . Then, the ego car transits into region \mathcal{R}_2 and approaches the front car which activates the safety constraint (23c). Consequently, the controller must reduce the velocity to keep the safe distance. Due to the adverse effect of the piecewise residuals, the Single GP-SOCP and nominal QP controller cannot maintain the safe distance. However, the proposed MOGP-SOCP controller effectively avoids this unsafe behavior by leveraging the batch MOGP design. A simulation video showing the performance of the nominal and MOGP-SOCP controller is provided. Figure 1 shows a snapshot of a simulation when all controllers result in the closest distance to the front car. It can be verified that only MOGP-SOCP controllers can keep the safe distance during the simulation.

In Figure 2, we aim to compare the performance of the proposed method with the baseline and nominal controllers. For your reference, we provided the trajectories of the oracle true design. It has been illustrated in Figure 2 that the single GP-SOCP controller did not converge to the desired velocity, while MOGP-SOCP and nominal QP controllers could reach v_d in the first 4 seconds. Then, the ego car transited to \mathcal{R}_2 at $t = 6 \text{ s}$ which affects the rate of change of the velocity. As it approaches the front car, the controllers must decrease the velocity in response to remain safe. The CBF $h(\mathbf{x})$ and control input u are also illustrated in this figure. This plot shows that the single GP-SOCP and nominal QP controllers violate the safety condition approximately at $t = 15 \text{ s}$ and $t = 17 \text{ s}$, respectively. However, the MOGP-SOCP controller could maintain the safe distance to the front car during the simulation. Also, it can be verified from the control input trajectory that the MOGP-SOCP controller could recover the true system's performance. However, the single GP-SOCP controller does not generate a smooth control input which is a result of poor uncertainty quantification between the switching times.

V. CONCLUSION

In this paper, A batch MOGP framework is developed to approximate the effect of the uncertainty on the CLF and CBF constraints. The switching dynamics of the true system result in piecewise residuals in CLF and CBF constraints corresponding to R regions that cover the state space. A batch MOGP model is designed to capture the piecewise

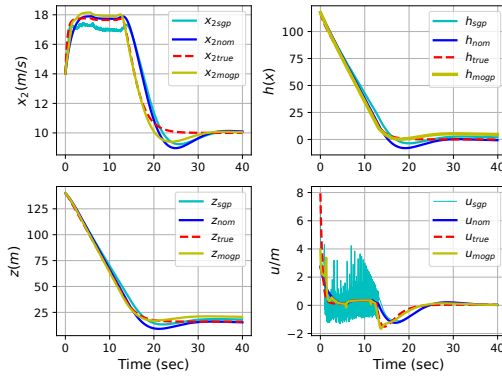


Fig. 2. Comparison of the MOGP-SOCP (yellow), single GP-SOCP (cyan), nominal QP (blue), and true design (red dashed). System state x_2 and distance between two cars z (left column). The CBF and control input u (right column).

residuals in each region. Then, the resulting constrained optimization problem with the uncertainty-aware chance constraint is converted into a SOCP. This optimization is proven to be convex and can be solved in real-time. Finally, the feasibility of the resulting SOCP is addressed. Future works will focus on eliminating the assumption of no impulse effects and non-overlapping regions in the switching system.

REFERENCES

- [1] Mohammad Aali and Jun Liu. Learning high-order control barrier functions for safety-critical control with gaussian processes. *arXiv preprint arXiv:2403.09573*, 2024.
- [2] Mohammad Aali and Jun Liu. Learning piecewise residuals of control barrier functions for safety of switching systems using multi-output gaussian processes. In *Proc. of ECC*, 2024.
- [3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proc. of ECC*, pages 3420–3431. IEEE, 2019.
- [4] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.
- [5] Franco Blanchini, Stefano Miani, et al. *Set-theoretic methods in control*, volume 78. Springer, 2008.
- [6] Stephen P Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [7] Michael S Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on automatic control*, 43(4):475–482, 1998.
- [8] Fernando Castañeda, Jason J Choi, Bike Zhang, Claire J Tomlin, and Koushil Sreenath. Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty. In *Proc. of CDC*, pages 6762–6769. IEEE, 2021.
- [9] Noel Csomay-Shanklin, Ryan K Cosner, Min Dai, Andrew J Taylor, and Aaron D Ames. Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety. In *Proc. of L4DC*, pages 1041–1053. PMLR, 2021.
- [10] Jamal Daafouz, Pierre Riedinger, and Claude Iung. Stability analysis and control synthesis for switched systems: a switched lyapunov function approach. *IEEE transactions on automatic control*, 47(11):1883–1887, 2002.
- [11] Leroy D’Souza and Yash Vardhan Pant. Stochastic hybrid model predictive control using gaussian processes for systems with piecewise residual dynamics. In *Proc. of ACC*, pages 2670–2677. IEEE, 2023.
- [12] David D Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Bayesian learning-based adaptive control for safety critical systems. In *Proc. of ICRA*, pages 4093–4099. IEEE, 2020.
- [13] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.
- [14] Jack K Hale. *Ordinary differential equations*. Courier Corp., 2009.
- [15] Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. In *Proc. of CDC*, pages 3699–3704. IEEE, 2020.
- [16] Hassan K Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- [17] Haitao Liu, Jianfei Cai, and Yew-Soon Ong. Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.
- [18] Jun Liu, Xinzhi Liu, and Wei-Chau Xie. Input-to-state stability of impulsive and switching hybrid systems with time-delay. *Automatica*, 47(5):899–908, 2011.
- [19] Mathias Marley, Roger Skjetne, and Andrew R Teel. Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles. In *Proc. of ACC*, pages 243–249. IEEE, 2021.
- [20] Yiming Meng and Jun Liu. Lyapunov-barrier characterization of robust reach-avoid-stay specifications for hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 49:101340, 2023.
- [21] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Proc. of HSCC*, pages 477–492. Springer, 2004.
- [22] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov-barrier function. *Automatica*, 66:39–47, 2016.
- [23] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.
- [24] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *Proc. of ICRA*, pages 2460–2465. IEEE, 2018.
- [25] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT Press, 2006.