

# Real-Time Adaptive Safety-Critical Control with Gaussian Processes in High-Order Uncertain Models

Yu Zhang<sup>1</sup>, Long Wen<sup>1</sup>, Xiangtong Yao<sup>1</sup>, Zhenshan Bing<sup>1\*</sup>, Linghuan Kong<sup>2</sup>,  
 Wei He<sup>3</sup>, and Alois Knoll<sup>1</sup>

**Abstract**—This paper presents an adaptive online learning framework for systems with uncertain parameters to ensure safety-critical control in non-stationary environments. Our approach consists of two phases. The initial phase is centered on a novel sparse Gaussian process (GP) framework. We first integrate a forgetting factor to refine a variational sparse GP algorithm, thus enhancing its adaptability. Subsequently, the hyperparameters of the Gaussian model are trained with a specially compound kernel, and the Gaussian model’s online inferential capability and computational efficiency are strengthened by updating a solitary inducing point derived from newly samples, in conjunction with the learned hyperparameters. In the second phase, we propose a safety filter based on high order control barrier functions (HOCBFs), synergized with the previously trained learning model. By leveraging the compound kernel from the first phase, we effectively address the inherent limitations of GPs in handling high-dimensional problems for real-time applications. The derived controller ensures a rigorous lower bound on the probability of satisfying the safety specification. Finally, the efficacy of our proposed algorithm is demonstrated through real-time obstacle avoidance experiments executed using both simulation platform and a real-world 7-DOF robot.

## I. INTRODUCTION

Recently, the increasing demand for safety-certified controllers in autonomous systems has been driven by the emergence of control barrier functions (CBFs). CBFs offer a framework for ensuring state forward-invariance, requiring just a single-step horizon for assurance [1]. This reduces computational effort, making it suitable for real-time applications, such as walking robot [2], autonomous driving [3], robotic manipulator [4], drone control [5], and multi-agent system [6]. Note that CBFs are often integrated into a quadratic programming (QP) problem as constraints, acting as safety filters to ensure system safety. This leads to a strong dependence on precise models, which are difficult to obtain in real-world applications [7]. Therefore, ensuring system safety in non-stationary environments with imprecise system models has become a recent research focus area.

System dynamics are typically viewed as a combination of the nominal and an uncertain component. This uncertainty arises from inaccuracies in the model parameters, and models with this kind of uncertainty are referred to as uncertain models. In scenarios where the nominal part closely approximates the real model, robust CBFs are suitable to be employed to

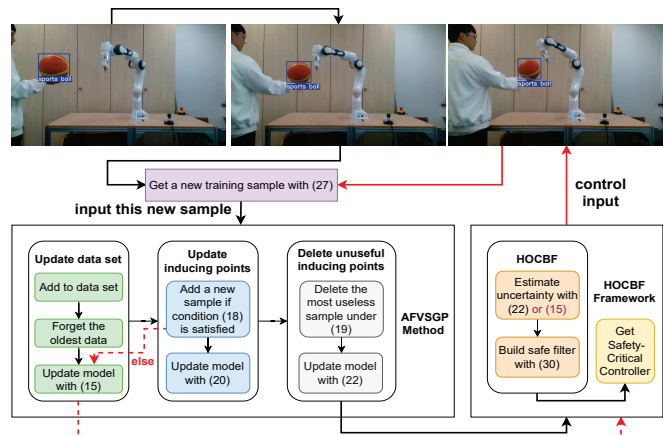


Fig. 1 An overview of the proposed AFVSGP-HOCBF framework. AFVSGP estimates the model-induced uncertainty affecting HOCBF, while HOCBF is applied to guarantee system safety.

compensate for the discrepancies [8]–[15]. However, when the disparity between the two becomes overly large, this method tends to be excessively conservative. For example, the robotic manipulator starts avoiding obstacles even when they are far away, significantly reducing its operational workspace for its inherent tasks. To address the issue of excessive conservatism, the focus has increasingly shifted towards learning-based methods. Among these, Gaussian processes (GPs) stand out because they offer a Bayesian, data-driven approach that requires minimal prior knowledge for accurately modeling complex functions of arbitrary complexity [16]. In [17], CBFs were firstly combined with GPs to devise controllers ensuring safety for affine systems with unknown drift term, and [18] further made a comparison between GPs and dropout neural networks [19], as well as ALPaCA [20], highlighting GPs’ superiority.

The early implementations of GP-CBFs for ensuring system safety exhibited several limitations. Firstly, offline learning strategies in GPs require extensive data collection to formulate a comprehensive global model, thereby reducing its adaptability in dynamic changes of the environment. Secondly, the inherent complexity of high-dimensional nonlinear models poses challenges for GPs, as a GP can only estimate a scalar function. An excessive number of GPs substantially diminishes the likelihood of maintaining system security. Lastly, introducing uncertainties in the input gain matrix of system dynamics violate the convexity in the QP problem, thereby increasing its solution time. Existing methods mostly address these drawbacks of GP-CBFs approaches through two primary strategies. The first strategy employs sparse online GP (SOGP) [21], to decrease the time complexity

<sup>1</sup>Y. Zhang, L. Wen, X. Yao, Z. Bing, and A. Knoll are with the Department of Informatics, Technical University of Munich, Germany. Corresponding author: Zhenshan Bing (zhenshan.bing@tum.de).

<sup>2</sup>L. Kong is with the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macau, China.

<sup>3</sup>W. He is with the School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, China, and also with the Key Laboratory of Intelligent Bionic Unmanned Systems, Ministry of Education, University of Science and Technology Beijing, Beijing 100083, China.

from  $\mathcal{O}(N^3)$  to  $\mathcal{O}(NM^2)$ , where  $N$  is the number of data and  $M$  represents the number of most representative points in the data, i.e.,  $M \ll N$ . The second strategy addresses the challenge of inferring control inputs by altering the kernel functions within the GPs. In [22], a novel matrix variate GP was introduced to simultaneously estimate the drift term and input gain of affine systems. The accompanying time complexity is  $\mathcal{O}(N^3)$  with learning hyperparameters and  $\mathcal{O}(NM^2)$  without learning hyperparameters. In [23], a GPs framework based on a compound kernel function was applied to ensure the GP-CBF-QP problem's convexity, providing both necessary and sufficient conditions for the feasible solution, with a time complexity of  $\mathcal{O}(N^3)$ .

However, to enable real-time safety in non-stationary environments using GP-CBFs control algorithms, three key challenges must still be addressed. Firstly, the algorithm should accommodate input gain uncertainties and be applicable to high-dimensional nonlinear models. Secondly, the adaptive selection of training data and inducing points must be considered to reflect the evolving nature of the environment. Lastly, optimizing time complexity in GPs is critical, especially for matrix inversion tasks during online model updates and inference. In this paper, we propose a novel adaptive online learning framework and the main contributions are outlined below. 1) A specific kernel function is introduced in the variational sparse GP (VSGP), enhancing online inference in high-dimensional nonlinear models with uncertain input gains, while maintaining VSGP's inherent benefit of reducing overfitting. 2) This is the first effort to incorporate novel forgetting mechanisms and rules in the VSGP-CBFs methodology. The goal is to make the model more adaptive to environmental changes by selecting the most informationally-rich training and inducing points. 3) The online update of the model's posterior distribution reduces the time complexity to  $\mathcal{O}(M^3)$ , as opposed to  $\mathcal{O}(NM^2)$  in the existing methods. 4) This work explores high order CBFs instead of conventional CBFs, providing a more comprehensive theoretical framework. The efficacy and real-time performance of the proposed algorithm were substantiated through dynamic obstacle avoidance tasks, executed both in the simulation platform and on a real-world 7-DOF robotic manipulator.

## II. BACKGROUND & PRELIMINARIES

In this section, we provide a brief introduction to the mathematical foundations of high order CBFs (HOCBF) [24] and VSGP [25], upon which our novel adaptive online learning framework builds.

### A. High Order Control Barrier Function

The paper focuses on a class of control-affine systems, as described below.

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \quad (1)$$

where  $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$  is the state;  $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$  is the control input;  $\mathcal{U}$  is a compact set;  $f(\mathbf{x})$  and  $g(\mathbf{x})$  are locally Lipschitz continuous and represent drift term and input gain, respectively. The manipulator  $\mathcal{A}$  is controlled by a Lipschitz continuous control law  $\mathbf{u} = k(\mathbf{x})$ .

We consider a closed convex set  $\mathcal{F}$  and represent it as the super level set of a continuously differentiable function

$h: \mathcal{X} \subset \mathbb{R}^n \subset \mathbb{R}$ , i.e.,

$$\begin{aligned} \mathcal{F} &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) \geq 0\} \\ \partial\mathcal{F} &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) = 0\} \\ \text{Int}(\mathcal{F}) &= \{x \in \mathcal{X} \subset \mathbb{R}^n : h(x) > 0\} \end{aligned} \quad (2)$$

where  $\mathcal{F}$  is the nonempty safe set, i.e.,  $\text{Int}(\mathcal{F}) \neq \emptyset$ , and we assume that there is no isolated point in  $\mathcal{F}$ .

*Definition 1:* (Invariant Set) Assuming that the initial conditions  $\mathbf{x}_0 \in \mathcal{F} \subset \mathcal{X}$  and subsequent states  $\mathbf{x} \in \mathcal{F} \subset \mathcal{X}$  for time  $t \in [t_0, t_{\max})$  can be achieved by designing an appropriate input  $\mathbf{u}(t)$ , we can assert that the set  $\mathcal{F}$  is a controlled invariant set for system (1). Additionally, system (1) is forward complete when  $t_{\max} = \infty$ .

The *Definition 1* dictates that for any state  $\mathbf{x} \in \mathcal{F}$ , it can be sustained within the set  $\mathcal{F}$  using permissible control inputs. Furthermore, for system (1), its forward completeness for  $t_{\max} = \infty$  suggests that, under the prescribed control input, the system's evolution can span an indefinite time horizon while staying inside the set  $\mathcal{F}$ .

*Definition 2:* (Relative degree) The relative degree of a control barrier function  $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$  with respect to the system (1) is defined as the minimum number of times the function needs to be differentiated along the dynamics of (1) until the control input  $\mathbf{u}$  appears explicitly.

*Definition 3:* (HOCBF) A function  $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  is a HOCBF of relative degree  $m$  for system if there exist  $m$ -ith order differentiable class  $\mathcal{K}$  functions  $\alpha_i$  and a class  $\mathcal{K}$  function  $a_m$ , such that:  $\sup_{\mathbf{u} \in \mathcal{U}} [L_f^m b(x) + L_g L_f^{m-1} b(x)\mathbf{u} + S(b(x)) + \alpha_m(\psi_{m-1}(x))] \geq 0$  for all  $x \in C_1 \cap \dots \cap C_m$ , where  $C_i = \{x \in \mathbb{R}^n : \psi_{i-1}(x) \geq 0\}$  for all  $i \in \{1, \dots, m\}$ ,  $\psi_i(x) = \psi_{i-1}(x) + \alpha_i(\psi_{i-1}(x))$ , and  $\psi_0(x) = h(x)$ .  $S(b_i(x))$  denotes the remaining Lie derivative along  $f$  with degree less than or equal to  $m-1$ , that is,  $S(b_i(x)) = \sum_{i=1}^{m-1} L_f^i (\alpha_{m-i} \circ \psi_{m-i-1})(x)$ .

*Proposition 1:* (Sufficient condition for safety) Consider the zero super-level set of a continuously differentiable control barrier function  $h(x)$ , denoted as  $\mathcal{F}$ . If the condition  $\mathbf{x}_0 \in C_1 \cap \dots \cap C_m$  holds, then the set  $C_1 \cap \dots \cap C_m$  is forward invariant, and the system can be guaranteed to operate safely via the implementation of a control policy, denoted as  $\pi(\mathbf{x})$ , that satisfies the constraint  $\pi(\mathbf{x}) \in \{\mathbf{u} \in \mathcal{U} \mid L_f^m b(x) + L_g L_f^{m-1} b(x)\mathbf{u} + S(b(x)) + \alpha_m(\psi_{m-1}(x)) \geq 0\}$ .

To identify a safety controller for system (1), the HOCBF is finally integrated with a QP problem as delineated below to find the minimum perturbation on  $u_{\text{nom}}$ .

$$\begin{aligned} u(x) &= \underset{\mathbf{u} \in \mathbb{R}^m}{\text{argmin}} \frac{1}{2} \|\mathbf{u} - u_{\text{nom}}\|^2 && \text{(HOCBF-QP)} \\ \text{s.t. } &L_f^m b(x) + L_g L_f^{m-1} b(x)\mathbf{u} + S(b(x)) && (3) \\ &+ \alpha_m(\psi_{m-1}(x)) \geq 0 \end{aligned}$$

where the term  $u_{\text{nom}}$  represents a nominal feedback controller. Note that  $u_{\text{nom}}$  can be derived through various methodologies including model predictive control, control Lyapunov function, neural network-based strategies, proportional derivative control, or reinforcement learning.

### B. Variational Sparse Gaussian Processes

The VSGP approach outlined in [25] employs inducing points as variational parameters to approximate the true GP posterior, thereby reducing overfitting and enabling the SGP

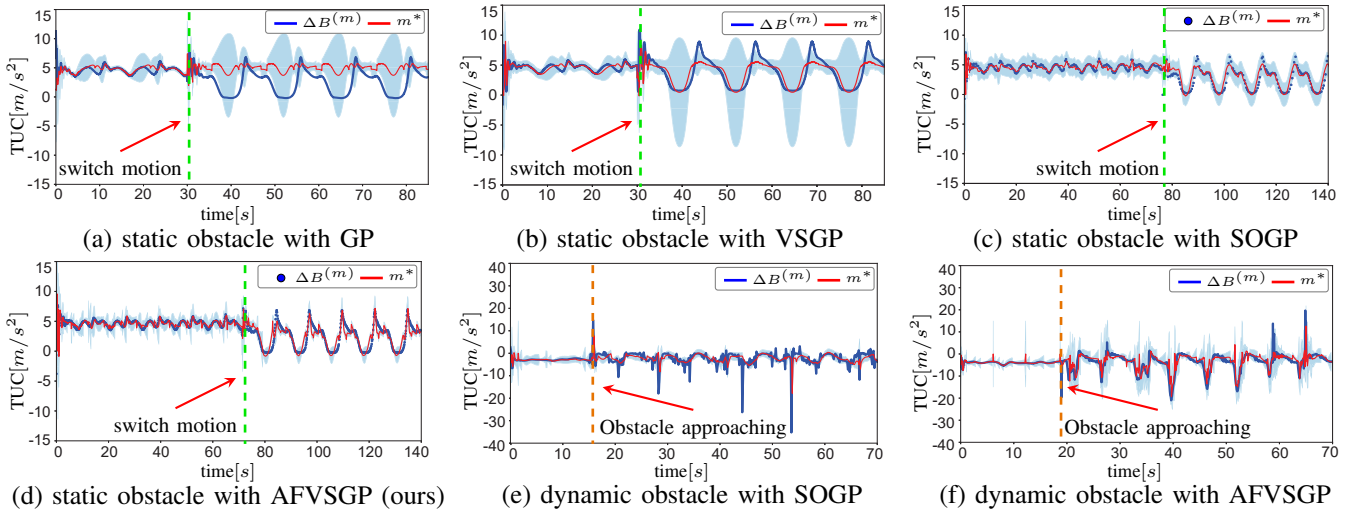


Fig. 2 Comparative analysis of predictive distributions across various learning methods under Gazebo simulation platform. Blue lines indicate the discrepancies between nominal estimates and actual model values, while red lines represent predictions from different models. A higher degree of alignment between red and blue lines suggests superior model performance. TUC represents the uncertain component with HOCBF and the blue shaded region represents the 95% confidence interval (the same applies to other figures).

to converge to its original form. Consider a data set  $(\xi, z)$ , which is derived from the relationship  $z_j = f(\xi_j) + \varepsilon_n$  where  $f : \mathbb{R}^b \rightarrow \mathbb{R}$ . The prior information is  $\xi := [\xi_1, \xi_2, \dots, \xi_N]^T \in \mathbb{R}^{N \times b}$ , while  $z$  is defined as  $z := [z_1, z_2, \dots, z_N]^T \in \mathbb{R}^N$ . Here,  $N$  denotes the number of training data points,  $b$  specifies the input dimensionality, and the term  $\varepsilon_n$ , which is distributed as  $\varepsilon_n \sim \mathcal{N}(0, \sigma_n^2)$ , accounts for Gaussian white measurement noise. Next, we will delve into the distinct features of this algorithm. A collection of inducing inputs are denoted by  $\mathbf{O} = [\mathbf{o}_1, \dots, \mathbf{o}_M]$ , where each  $\mathbf{o}_i$  belongs to  $\xi$ . These inputs correspond to latent variables  $\mathbf{f}_o = [f(\mathbf{o}_1), \dots, f(\mathbf{o}_M)]^T$ . Under the assumption that  $\mathbf{f}_o$  serves as a sufficient statistic for  $\mathbf{f}$ , the predictive posterior of the VSGP, given by (4), can be approximated using (5), as detailed subsequently.

$$p(f_* | z) = \iint p(f_* | \mathbf{f}_o, \mathbf{f}) p(\mathbf{f} | \mathbf{f}_o, z) p(\mathbf{f}_o | z) d\mathbf{f} d\mathbf{f}_o \quad (4)$$

$$q(f_*) = \int p(f_* | \mathbf{f}_o) q(\mathbf{f}_o) d\mathbf{f}_o \quad (5)$$

where  $p(f_* | \mathbf{f}_o, \mathbf{f})$  is approximated by  $p(f_* | \mathbf{f}_o)$ , and posterior distribution  $p(\mathbf{f}_o | z)$  is approximated by a gaussian variational distribution  $q(\mathbf{f}_o)$  with mean  $\boldsymbol{\mu}$  and covariance  $\mathbf{A}$ , and the approximate predictive posterior is shown as

$$\begin{aligned} q(f_*) &= \mathcal{N}(f_* | m_*, v_*) \\ m_* &= \mathbf{k}_{*o} \mathbf{K}_{oo}^{-1} \boldsymbol{\mu} \\ v_* &= k_{**} - \mathbf{k}_{*o} \mathbf{K}_{oo}^{-1} \mathbf{k}_{o*} + \mathbf{k}_{*o} \mathbf{K}_{oo}^{-1} \mathbf{A} \mathbf{K}_{oo}^{-1} \mathbf{k}_{o*}, \end{aligned} \quad (6)$$

where  $\mathbf{k}_{*o} = \mathbf{k}_{o*}^T$  is a row vector with the kernel products between  $\xi^*$  and  $\mathbf{O}$ , and covariance  $\mathbf{K}_{oo}$  is the matrix that contain all kernel dot products  $k(\mathbf{o}, \mathbf{o}')$  for inducing inputs, and  $k_{**} = k(\xi_*, \xi_*)$  is the kernel product of a query point.

To obtain the variational parameters  $\boldsymbol{\mu}$  and  $\mathbf{A}$ , authors in [25] firstly introduced the subsequent variational lower bound

for the authentic log marginal likelihood.

$$\log p(\mathbf{z}) \geq F_V(\mathbf{O}, q(\mathbf{f}_o)) = \int q(\mathbf{f}_o) \log \frac{G(\mathbf{f}_o, \mathbf{z}) p(\mathbf{f}_o)}{q(\mathbf{f}_o)} d\mathbf{f}_o \quad (7)$$

where the link  $\log G(\mathbf{f}_o, \mathbf{z}) = \log \mathcal{N}(\mathbf{z} | \mathbf{K}_{\xi o} \mathbf{K}_{oo}^{-1} \mathbf{f}_o, \sigma^2 \mathbf{I}) - (1/2\sigma^2) \text{tr}\{\mathbf{K}_{\xi\xi} - \mathbf{K}_{\xi o} \mathbf{K}_{oo}^{-1} \mathbf{K}_{o\xi}\}$  is shown with the trace operator  $\text{tr}\{\cdot\}$ . Subsequently, authors differentiate (7) with respect to  $q(\mathbf{f}_o)$  and equate it to zero to obtain the optimal expression shown as follows.

$$q(\mathbf{f}_o) = \mathcal{N}(\mathbf{f}_o | \boldsymbol{\mu}, \mathbf{A}) \quad (8)$$

where  $\boldsymbol{\mu} = \sigma^{-2} \mathbf{K}_{oo} \mathbf{B} \mathbf{K}_{o\xi} \mathbf{z}$ ,  $\mathbf{A} = \mathbf{K}_{oo} \mathbf{B} \mathbf{K}_{oo}$ ,  $\mathbf{B} = (\mathbf{K}_{oo} + \sigma^{-2} \mathbf{K}_{o\xi} \mathbf{K}_{\xi o})^{-1}$ . To further refine the hyperparameters, including the noise variance and the kernel parameters, and to update the set of  $\mathbf{O}$ , the methodology outlined in [25] utilized Jensen's inequality. This was employed to eliminate the distribution  $q(\mathbf{f}_o)$ , resulting in a collapsed version of the bound presented in (7).

$$\begin{aligned} F_V(\mathbf{O}) &= \log \mathcal{N}(\mathbf{z} | \mathbf{0}, \sigma^2 \mathbf{I} + \mathbf{K}_{\xi o} \mathbf{K}_{oo}^{-1} \mathbf{K}_{o\xi}) \\ &\quad - \frac{1}{2\sigma^2} \text{tr}\{\mathbf{K}_{\xi\xi} - \mathbf{K}_{\xi o} \mathbf{K}_{oo}^{-1} \mathbf{K}_{o\xi}\} \end{aligned} \quad (9)$$

Ultimately, the optimal values are obtained by maximizing the bound in (9), and both the computational and optimization burden for (8) and (9) are  $\mathcal{O}(NM^2)$ , respectively.

### III. ADAPTIVE ONLINE VSGP WITH THE AFFINE DOT PRODUCT KERNEL

This section comprises three key components for our proposed novel adaptive fast VSGP (AFVSGP) framework: 1) offline hyperparameter training, 2) adaptive modification and forgetting of training data and inducing points, and 3) real-time model inference updates.

#### A. Offline Training for Hyperparameter

We begin by selecting a small subset of preliminary data, denoted by  $P$  for offline hyperparameter training, where

$P \ll N$  and  $N$  is the typical training data size in standard GPs. Following this, we integrate a specialized kernel function [26] into the VSGP algorithm.

*Definition 4:* Affine Dot Product compound kernel: Define  $k_a : \bar{\mathcal{X}} \times \bar{\mathcal{X}} \rightarrow \mathbb{R}$  given by

$$k_a \left( \begin{bmatrix} \xi \\ \bar{u} \end{bmatrix}, \begin{bmatrix} \xi' \\ \bar{u}' \end{bmatrix} \right) = \bar{u}^T \text{Diag} (k_1(\xi, \xi'), \dots, k_{m+1}(\xi, \xi')) \bar{u}' \quad (10)$$

as the Affine Dot Product (ADP) compound kernel of  $(m+1)$  individual kernels  $k_1, \dots, k_{m+1} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ .

*Proposition 2:* Let  $k_1, \dots, k_{m+1}$  be a set of positive definite kernels. Then the ADP compound kernel  $k_a$  inherits positive definiteness. Additionally, if  $k_1, \dots, k_{m+1}$  are bounded,  $k_a$  remains bounded.

Building upon Thm. 1 in [26],  $\mathcal{H}_{k_a}(\xi \times \bar{U})$  is an Reproducing Kernel Hilbert Space (RKHS) whose reproducing kernel is  $k_a$  in Definition 4. We include a forgetting factor  $\phi$  into this term and start considering time instant  $t$ , so only data from initial sample at  $t' = 1$  to  $t$  are available, then, with the ADP kernel, (6) is rewritten as

$$\begin{aligned} q(f_*) &= \mathcal{N}(f_* | m_{*,\phi}, v_{*,\phi}) \\ m_{*,\phi} &= \sigma^{-2} \mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o} \mathbf{B}_\phi (\mathbf{k}_{*\bar{U}_o}^\top) \bar{u}_* \\ v_{*,\phi} &= \bar{u}_*^\top (k_{**} + \mathbf{k}_{*\bar{U}_o} (\mathbf{B}_\phi - \mathbf{K}_{oo}^{-1}) \mathbf{k}_{\bar{U}_o*}) \bar{u}_* \end{aligned} \quad (11)$$

where  $\mathbf{z}_j = \mathcal{F}(\xi_j, \bar{u}_j) + \epsilon_n$ ,  $\mathbf{K}_{\xi_o} \in \mathbb{R}^{P \times M}$  is the gram matrix of  $\bar{u}_o^\top \text{Diag}([k_1(\xi, \xi'), \dots, k_{m+1}(\xi, \xi')]) \bar{u}'_o$ ,  $\mathbf{B}_\phi = (\mathbf{K}_{oo} + \sigma^{-2} \mathbf{K}_{o\xi} \mathbf{K}_{\xi_o})^{-1} \in \mathbb{R}^{M \times M}$  and  $\mathbf{K}_{oo}$  are also gram matrix of  $\bar{u}_o^\top \text{Diag}([k_1(\xi_o, \xi'_o), \dots, k_{m+1}(\xi_o, \xi'_o)]) \bar{u}'_o$ ,  $\mathbf{z} \in \mathbb{R}^P$ , and  $(\mathbf{k}_{*\bar{U}_o}) \in \mathbb{R}^{(m+1) \times M}$  is given by

$$\mathbf{k}_{*\bar{U}_o} = [K_{1*} \ K_{2*} \ \dots \ K_{(m+1)*}]^\top \circ * \bar{U}_o \quad (12)$$

$$K_{i*} = [k_i(\xi_*, \xi_1), \dots, k_i(\xi_*, \xi_M)] \quad (13)$$

and  $k_{**} = \text{Diag}\{k_1(\xi_*, \xi_*), \dots, k_{m+1}(\xi_*, \xi_*)\}$ , and where  $\mathbf{\Lambda}$  is a  $P \times P$  diagonal matrix with  $\Lambda_{t't'} = \phi^{P-t'}$ . To enhance clarity and promote an effective understanding of the text, we opt to utilize the symbol  $k(\cdot)$  in lieu of  $k_a(\cdot)$  in (10). To update hyperparameters, equation (9) is rewritten as.

$$\begin{aligned} F_V^\phi(\mathbf{O}) &\propto \log \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{K}_{\xi_o} \mathbf{K}_{oo}^{-1} \mathbf{K}_{o\xi} + \sigma^2 \mathbf{\Lambda}^{-1}) \\ &\quad - \frac{1}{2\sigma^2} \sum_{t'=1}^t (\phi^{t-t'} - 1) \log(2\pi\sigma^2) \\ &\quad + \frac{1}{2\sigma^2} \sum_{t'=1}^t \phi^{t-t'} (k_{t't'} - \mathbf{k}_{t'o} \mathbf{K}_{oo}^{-1} \mathbf{k}_{ot'}) \end{aligned} \quad (14)$$

where  $\mathbf{z} = [z_1, \dots, z_t]^\top$  and each kernel is replaced by the ADP compound kernel. Employing (11) and (14), we are capable of learning from the collected  $P$  data offline, simultaneously acquiring the hyperparameters and the approximate predictive posterior. This offline computational process carries a computational burden of  $\mathcal{O}(PM^2)$ .

### B. Adaptive Online Learning for Data and Inducing Points

Following offline analysis and hyperparameter optimization, we transition to online adaptive updates. This is essential for real-time posterior approximation upon the arrival of each new training sample  $(\xi_{t+1}, \bar{u}_{t+1}, z_{t+1})$ . The oldest

### Algorithm 1 AFVSGP-HOCBF-SOCP

- 1: **Inputs:** New sample  $((\xi_{t+1}, \bar{u}_{t+1}), z_{t+1})$ , offline model  $q(f_*)$  with (11), window  $P$ , thresholds  $R_{th}$ ,  $R_{thm}$ .
- 2: Update training set and compute  $(m_{*,\phi})_{t+1}$  and  $(v_{*,\phi})_{t+1}$  with (15) for new data sample.
- 3: **if**  $P_{th} > \epsilon$  [see (16)] **then**
- 4:   set the new sample in inducing points as  $\mathbf{o}_{k+1}$  and update the predictive distribution with (18).
- 5: **end if**
- 6: **if**  $\mathbf{o}_{k-1}$  with less information [see (17)] **then**
- 7:   Update the predictive distribution with (20) and get the new model  $q(f_*)$ .
- 8: **end if**
- 9: Compute the safety-aware controller with (28) and get new sample with (25).
- 10: **return** Safety control input  $\bar{u}_*$ .

TABLE I Comparison of different methods while avoiding a static obstacle under Gazebo simulation platform.

| Model       | AFVSGP         | VSGP    | GP     | SOGP    |
|-------------|----------------|---------|--------|---------|
| MSE         | <b>0.5863</b>  | 1.4844  | 6.6029 | 0.8118  |
| Tr. Time(s) | <b>0.00613</b> | 0.03562 | /      | 0.08724 |

sample is replaced to maintain a fixed-size training set  $P$ , and both the mean and variance are updated at  $t+1$ .

$$\begin{aligned} (m_{*,\phi})_{t+1} &= \sigma^{-2} (\mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o})_{t+1} (\mathbf{B}_\phi)_{t+1} (\mathbf{k}_{*\bar{U}_o}^\top) \bar{u}_* \\ (v_{*,\phi})_{t+1} &= \bar{u}_*^\top (k_{**} + \mathbf{k}_{*\bar{U}_o} ((\mathbf{B}_\phi)_{t+1} - \mathbf{K}_{oo}^{-1}) \mathbf{k}_{\bar{U}_o*}) \bar{u}_* \end{aligned} \quad (15)$$

where

$$\begin{aligned} (\mathbf{B}_\phi)_{t+1} &= ((1-\phi)\mathbf{K}_{oo} + \phi(\mathbf{B}_\phi)_t^{-1} + \sigma^{-2} \mathbf{k}_{t+1,o}^\top \mathbf{k}_{t+1,o} \\ &\quad - \phi^\top \mathbf{k}_{t-T,o}^\top \mathbf{k}_{t-T,o})^{-1} \end{aligned}$$

$$(\mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o})_{t+1} = \phi (\mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o})_t + z_{t+1} \mathbf{k}_{t+1,o} - z_{t-T} \phi \mathbf{k}_{t-T,o}$$

If the previous iteration  $\mathbf{B}_\phi$  and  $\mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o}$  are saved, the complexity of operations  $(\mathbf{B}_\phi)_{t+1}$  and  $(\mathbf{z}^\top \mathbf{\Lambda} \mathbf{K}_{\xi_o})_{t+1}$  are  $\mathcal{O}(M^3)$  and  $\mathcal{O}(M)$ , respectively. After revising the training set, we use established metrics to select the most representative inducing points, as inspired by [27]–[29]. These points are added to the inducing set, while the least relevant ones are removed. The value of the top representative inducing point should exceed the subsequent regularization term linked to the adaptive collapsed marginal likelihood, detailed in (14).

$$P_{th} = \sum_{t'=1}^t \phi^{t-t'} (k_{t't'} - \mathbf{k}_{t'o} \mathbf{K}_{oo}^{-1} \mathbf{k}_{ot'}) \quad (16)$$

Thus, if a newly introduced training point results in a value from (16) exceeding a user-defined threshold  $\epsilon$ , i.e.,  $P_{th} > \epsilon$ , it is added to the inducing set. Additionally, we use the same equation to assess the representativeness of existing points. The least representative point is identified and removed based on criteria for each  $P_{thm}$ ,  $m = 1, \dots, M$ .

$$P_{thm} = \sum_{t'=1}^t \phi^{t-t'} \mathbf{k}_{t'm} \mathbf{K}_{mm}^{-1} \mathbf{k}_{mt'} \quad (17)$$

where  $P_{th} = \sum_{t'=1}^t \phi^{t-t'} k_{t't'} - \sum_{m=1}^M P_{thm}$ . On incorporating the new sample  $((\xi_{k+1}, \bar{u}_{k+1}), z_{k+1})$  as  $\mathbf{o}_{k+1}$  into

the inducing inputs  $\mathbf{O} = [\mathbf{o}_1, \dots, \mathbf{o}_M]$ , the approximate predictive posterior is updated using the following procedure.

$$\begin{aligned} (m_{*,\phi})_{k+1} &= \sigma^{-2} \mathbf{z}^\top \mathbf{\Lambda} (\mathbf{K}_{\xi_o})_{k+1} (\mathbf{B}_\phi)_{k+1} (\mathbf{k}_{*\bar{U}_o}^\top)_{k+1} \bar{u}_* \\ (v_{*,\phi})_{k+1} &= \bar{u}_*^\top (k_{**} + (\mathbf{k}_{*\bar{U}_o})_{k+1} ((\mathbf{B}_\phi)_{k+1} \\ &\quad - (\mathbf{K}_{oo}^{-1})_{k+1}) (\mathbf{k}_{\bar{U}_o*})_{k+1}) \bar{u}_*, \end{aligned} \quad (18)$$

where  $(\mathbf{k}_{*\bar{U}_o}^\top)_{k+1}$  is straightforward since it only implies adding a row and new element to this vector and matrix, respectively. The update of  $(\mathbf{B}_\phi)_{k+1}$  can be effectively computed from  $(\mathbf{B}_\phi)_k$  using the properties of the block matrix inversion.

$$(\mathbf{B}_\phi)_{k+1} = \begin{pmatrix} (\mathbf{B}_\phi^{-1})_k & (\mathbf{b}_\phi)_{k+1} \\ (\mathbf{b}_\phi^\top)_{k+1} & (b_\phi)_{k+1} \end{pmatrix}^{-1} \quad (19)$$

where  $(b_\phi)_{k+1} = k_{k+1,k+1} + \sigma^{-2} \mathbf{k}_{k+1,\xi} \mathbf{\Lambda} \mathbf{k}_{\xi,k+1}$  and  $(\mathbf{b}_\phi)_{k+1} = \mathbf{k}_{o,k+1} + \sigma^{-2} (\mathbf{K}_{o\xi})_k \mathbf{\Lambda} \mathbf{k}_{\xi,k+1}$ . Consequently, employing (18) and (19), the time complexity associated with the addition of a new inducing point is found to be  $\mathcal{O}(M^3)$  due to compute the inversions of  $(\mathbf{B}_\phi)_k$  and  $(\mathbf{K}_{oo})_k^{-1}$ . In addition, when we delete a inducing point in the set, the mean and variance is rewritten as

$$\begin{aligned} (m_{*,\phi})_{k-1} &= \sigma^{-2} \mathbf{z}^\top \mathbf{\Lambda} (\mathbf{K}_{\xi_o})_{k-1} (\mathbf{B}_\phi)_{k-1} (\mathbf{k}_{*\bar{U}_o}^\top)_{k-1} \bar{u}_* \\ (v_{*,\phi})_{k-1} &= \bar{u}_*^\top (k_{**} + (\mathbf{k}_{*\bar{U}_o})_{k-1} ((\mathbf{B}_\phi)_{k-1} \\ &\quad - (\mathbf{K}_{oo}^{-1})_{k-1}) (\mathbf{k}_{\bar{U}_o*})_{k-1}) \bar{u}_* \end{aligned} \quad (20)$$

where  $(\mathbf{B}_\phi)_{k-1} = ((\mathbf{K}_{oo})_{k-1} + \sigma^{-2} (\mathbf{K}_{o\xi})_{k-1} \mathbf{\Lambda} (\mathbf{K}_{\xi_o})_{k-1})^{-1}$ . Other  $(k-1)$  elements are delating a row, a column, or an element to its vector and matrix, so we omit its process.

#### IV. PROPOSED SAFETY CONTROLLER

In this section, we aim to integrate this AFVSGP framework into (3) to enhance safety. The  $m$ th derivatives of the HOCBF for the affine system can be expressed as

$$B^{(m)}(\xi, \bar{u}) = \mathcal{F}(\xi) \bar{u} + S(b(\xi)) + \alpha_m (\psi_{m-1}(\xi)) \quad (21)$$

where  $\mathcal{F}(\xi) = [L_f^m b(\xi) \quad L_g L_f^{m-1} b(\xi)]$  and  $\bar{u} = [1 \quad u^\top]^\top$ .

The nominal estimates of their values are given by

$$B_0^{(m)}(\xi, \bar{u}) = \mathcal{F}_0(\xi) \bar{u} + S(b(\xi)) + \alpha_m (\psi_{m-1}(\xi)) \quad (22)$$

where  $\mathcal{F}(\xi) = [L_{f0}^m b(\xi) \quad L_{g0} L_{f0}^{m-1} b(x)]$ .

We introduce the function  $\Delta B^{(m)} : \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}$  to represent the discrepancy between (21) and (22).

$$\Delta B^{(m)}(\xi, \bar{u}) := B^{(m)}(\xi, \bar{u}) - B_0^{(m)}(\xi, \bar{u}). \quad (23)$$

Then, the HOCBF constraints become

$$B_0^{(m)}(\xi, \bar{u}) + \Delta B^{(m)}(\xi, \bar{u}) \geq 0 \quad (24)$$

It should be noted that the uncertainties,  $\Delta B^{(m)}(\xi, \bar{u})$ , are scalar values. This results in a dimensionally reduced learning problem compared to directly learning the dynamics,  $\mathcal{F}(\xi)$ . To approximate  $\Delta B^{(m)}(\xi, \bar{u})$ , we can collect trajectories from the actual plant. These measurements are provided as follows.

$$\begin{aligned} z_j^{\Delta B^{(m)}} &= (B^{(m-1)}(\xi(t + \Delta t)) - B^{(m-1)}(\xi(t))) / \Delta t \\ &\quad - B_0^{(m)}(\xi_j, u_j) \end{aligned} \quad (25)$$

where  $\xi_j = (\xi(t + \Delta t) + \xi(t))/2$  represents the state's average over the interval  $(t, t + \Delta t)$ . The term  $u_j$  denotes the control input within the same timeframe. The parameter  $\Delta t$  defines the sampling interval. Meanwhile,  $z_j^{\Delta B^{(m)}}$  serves as an approximation of the measurements for  $\Delta B^{(m)}(\xi, \bar{u})$  for  $j = 1, \dots, N$ . Given these datasets, the regression challenges concerning  $\Delta B^{(m)}(\xi, \bar{u})$  can be effectively framed as supervised learning tasks.

*Assumption 1:* The candidate function  $\Delta B^{(m)}(\xi, \bar{u})$  belong to the RKHS [30], associated to the reproducing kernel function (10). Therefore, it has bounded RKHS norm with respect to  $k_a$ , i.e.,  $\|\Delta B^{(m)}(\xi, \bar{u})\|_{k_a}^2 \leq \|\Delta\|_{k_a}$ .

According to Thm. 2 in [26], each subsequent inequality is satisfied with a probability of at least  $1 - \delta$  for all  $\xi_i \in \xi$  and  $\bar{u} \in \bar{U}$ .

$$B^{(m)}(\xi, \bar{u}) \geq B_0^{(m)}(\xi, \bar{u}) + \mu_{\Delta B}(\xi, \bar{u}) - \beta \sigma_{\Delta B}(\xi, \bar{u}) \quad (26)$$

where  $\beta := (2\|\Delta\|_{k_a}^2 + 300\kappa_{P+1} \ln^3((P+1)/\delta))^{0.5}$ , with  $P$  is the total number of data,  $\delta \in (0, 1)$ ,  $\kappa_{P+1}$  represents the maximum information for which a uniform bound can be established, as detailed in [31], after obtaining  $P+1$  data points.  $\mu_{\Delta B}$  and  $\sigma_{\Delta B}$  are the mean and standard deviation of the GP prediction of  $\Delta B^{(m)}(\xi_*, \bar{u}_*)$ , given by (20).

We provide a detailed expression for  $\mu_{\Delta B}$  and  $\sigma_{\Delta B}$  as follows.

$$\begin{aligned} \mu_{\Delta B}(\xi, u | \mathbb{D}_N) &= b_{\Delta B}^\top(\xi | \mathbb{D}_N) [1 \quad u^\top]^\top \\ \sigma_{\Delta B}(\xi, u | \mathbb{D}_N) &= [1 \quad u^\top]^\top \Sigma_{\Delta B}(\xi | \mathbb{D}_N) [1 \quad u^\top]^\top \end{aligned} \quad (27)$$

where  $b_{\Delta B}^\top = \sigma^{-2} \mathbf{z}^\top \mathbf{\Lambda} (\mathbf{K}_{\xi_o})_{k-1} (\mathbf{B}_\phi)_{k-1} (\mathbf{k}_{*\bar{U}_o}^\top)_{k-1}$  and  $\Sigma_{\Delta B} = k_{**} + (\mathbf{k}_{*\bar{U}_o})_{k-1} ((\mathbf{B}_\phi)_{k-1} - (\mathbf{K}_{oo}^{-1})_{k-1}) (\mathbf{k}_{\bar{U}_o*})_{k-1}$ . The uncertainty-aware control for safety of system (1) is finally transformed as an AFVSGP-HOCBF-SOCP problem shown as follows, and detailed in Alg. 1 and Fig. 1.

$$\begin{aligned} u(x) &= \operatorname{argmin}_{u \in \mathbb{R}^m} \frac{1}{2} \|\bar{u} - \bar{u}_{\text{nom}}\|^2 \\ \text{s.t. } &\mathcal{F}_0(\xi) \bar{u} + S(b(\xi)) + \alpha_m (\psi_{m-1}(\xi)) + b_{\Delta B}^\top \bar{u} \\ &\quad - \beta \bar{u}^\top \Sigma_{\Delta B} \bar{u} \geq 0 \end{aligned} \quad (28)$$

Equation (28) adheres to the characteristics of a Second-Order Cone Program (SOCP). Numerous studies have emphasized its local Lipschitz continuity, underscoring the convex nature of the problem in data-driven safety-critical control when implementing GP-CBFs [26], [32], [33]. Moreover, once optimal hyperparameters have been determined through offline learning, our proposed framework guarantees an online learning time complexity of  $\mathcal{O}(M^3)$ , rendering the method highly applicable for real-time scenarios.

#### V. SIMULATION AND REAL-WORLD EXPERIMENTS

In this section, we evaluate obstacle avoidance tasks for a Franka robot in both Gazebo platform and real-world experiments. The Franka robot is with unknown parameters of inertial matrix, gravitational forces, coriolis and centrifugal matrix, and utilizing ROS for module's communication. Therefore, we set (22) equal to zero in this section. We adapt existing VSGP code [27] and customize kernel functions via Pyro [34]. Codes run on a 2.30GHz Core i7 CPU with 16GB RAM. For static obstacles, models are first trained and evaluated using data collected along the y-axis with the same

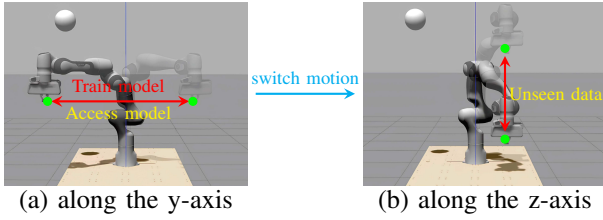


Fig. 3 Methods are initially trained and assessed using the end-effector's data on the y-axis. Subsequently, the motion is shifted to the z-axis to evaluate adaptability with previously unseen data.

motion scheme. After a predefined period, the evaluation is switched to the z-axis to assess the model's adaptability with unseen data, as depicted in Fig. 3. In this process, the uncertain component with HOCBF abbreviates as TUC. We optimize hyperparameters for the AFVSGP approach (ours) with only a 200-point dataset and 20 inducing points. By contrast, non-adaptive VSGP [25], conventional offline GP [23], and SOGP methods [21] use a 2000-point dataset for one cycle offline training along y-axis. For dynamic obstacles, the end-effector moves cyclically along the y-axis, and its efficacy is compared to the SOGP method.

For static obstacles, Table I shows the mean squared error (MSE) and online training time (Tr. Time) across different methods, while Fig. 2(d) confirms AFVSGP's ability for real-time adaptive learning. In contrast, VSGP, depicted in Fig. 2(b), lacks a forgetting mechanism, which renders it susceptible to the influence of prior trajectory data, consequently biasing its real-time learning. The offline GP algorithm, presented in Fig. 2(a), inadequately adapts to new trajectory data, resulting in suboptimal data fitting post-trajectory change. Figs. 2(c) and 2(d) scrutinize the volume of data collected during the online learning process. Our algorithm's lower computational complexity enables real-time collection of a more extensive dataset, yielding higher data density than the SOGP method [21].

For dynamic obstacle avoidance, as depicted in Fig. 2(f), our algorithm achieves a close fit with real-time data. Conversely, the SOGP-based method, illustrated in Fig. 2(e), struggles to incorporate sufficient real-time data effectively due to prolonged training durations. Fig. 4(b) illustrates that our algorithm successfully maintains a positive CBF domain to ensure safety. In contrast, the SOGP methods, as shown in Fig. 4(a), result in a minimal distance between the end-effector and the unsafe area border that is less than zero, leading to frequent collisions.

In the real-world experiment, the experimental platform utilizes a Franka robot equipped with two Intel D415 depth cameras for spatial localization. Obstacle classification is performed using the YOLOv8n model [35], as shown in Fig. 1. To safeguard the robotic manipulator and personnel, the experiments exclude failed simulation methods, focusing solely on known-parameter methods and AFVSGP-HOCBF (ours) with fully unknown parameters. The experimental design is segmented into three parts. The first part involves a comparative analysis of data fitting with and without learning in the presence of static obstacles. The second part focuses on complete online learning results under dynamic obstacle conditions. Finally, the third part contrasts the performance with known-parameter approaches. Following an abrupt switch to our algorithm, as shown in Fig. 5(a), the proposed method

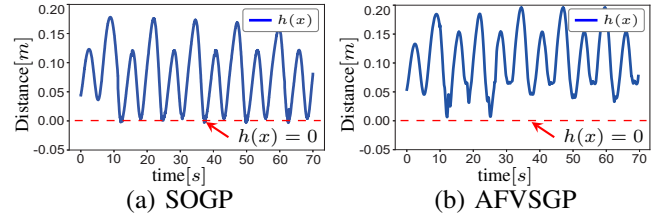


Fig. 4 Control barrier function across different learning methods during dynamic obstacle avoidance. Here,  $h(x)$  represents the minimum distance to the unsafe area's boundary.

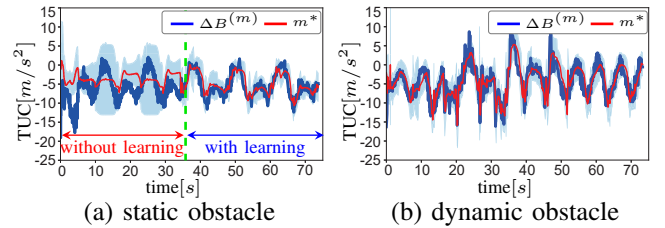


Fig. 5 Online learning outcomes for real-world obstacle avoidance with the Franka robot using the AFVSGP-HOCBF method.

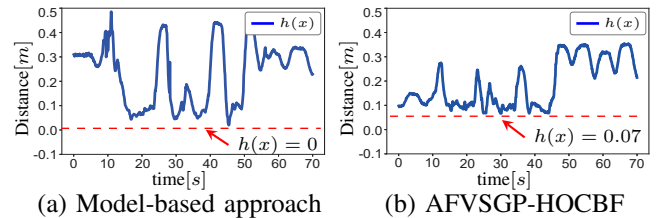


Fig. 6 Comparison of control barrier function between known-parameter methods and our AFVSGP-HOCBF approach with fully unknown parameters in real-world tests.

rapidly adapts to the data in the presence of static obstacles. In Fig. 5(b), due to its rapid inference capabilities, the proposed algorithm also effectively fits the required data in the presence of dynamic obstacles. In Fig. 6(b), the data shows a distance greater than 0.07m from the origin, indicating that the minimal distance between end-effector and the border line of unsafe area is 0.07m. This behavior is little conservative compared to model-based approaches, as shown in Fig. 6(a), a result of the confidence intervals inherent to our method. Nonetheless, our method ensures safety while maintaining an acceptable level of conservatism.

## VI. CONCLUSION

This paper introduces a novel adaptive online learning framework to address real-time safety concerns in higher-order nonlinear systems operating under model uncertainty and in non-stable environments. In the process of designing rules to select the most representative data points and reconstructing GPs model, the time complexity for updating prediction has been reduced from  $\mathcal{O}(NM^2)$  to  $\mathcal{O}(M^3)$ . This ensures real-time applicability in high-frequency systems. The feasibility of the algorithm in real-world scenarios is substantiated through empirical experiments. For future work, we will integrate additional factors into the proposed framework based on CBF, which is explored to address the challenges posed by multi-obstacle avoidance scenarios.

## REFERENCES

- [1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, pp. 3420–3431, 2019.
- [2] Q. Nguyen and K. Sreenath, "Safety-critical control for dynamic bipedal walking with precise footstep placement," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 147–154, 2015.
- [3] T. D. Son and Q. Nguyen, "Safety-critical control for non-affine nonlinear systems with application on autonomous vehicle," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 7623–7628, 2019.
- [4] Z. Bing, L. Knak, L. Cheng, F. O. Morin, K. Huang, and A. Knoll, "Meta-reinforcement learning in nonstationary and nonparametric environments," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.
- [5] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2460–2465, 2018.
- [6] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2021.
- [7] X. Yao, Z. Bing, G. Zhuang, K. Chen, H. Zhou, K. Huang, and A. Knoll, "Learning from symmetry: Meta-reinforcement learning with symmetrical behaviors and language instructions," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5574–5581, 2023.
- [8] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *2020 American Control Conference (ACC)*, pp. 1399–1405, 2020.
- [9] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1073–1088, 2022.
- [10] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Robust adaptive control barrier functions: An adaptive and data-driven approach to safety," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1031–1036, 2021.
- [11] J. Dong, W. Si, and C. Yang, "A novel human-robot skill transfer method for contact-rich manipulation task," *Robotics Intelligence and Automation*, vol. 43, no. 3, pp. 327–337, 2023.
- [12] Z. Bing, H. Zhou, R. Li, X. Su, F. O. Morin, K. Huang, and A. Knoll, "Solving robotic manipulation with sparse reward reinforcement learning via graph-based diversity and proximity," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 3, pp. 2759–2769, 2023.
- [13] T. Beckers, J. Umlauf, and S. Hirche, "Stable model-based control with gaussian process regression for robot manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3877–3884, 2017.
- [14] Z. Bing, A. Koch, X. Yao, K. Huang, and A. Knoll, "Meta-reinforcement learning via language instructions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5985–5991, 2023.
- [15] Z. Bing, M. Brucker, F. O. Morin, R. Li, X. Su, K. Huang, and A. Knoll, "Complex robotic manipulation via graph-based hindsight goal generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7863–7876, 2022.
- [16] J. Kocijan, *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.
- [17] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using gaussian processes," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3699–3704, 2020.
- [18] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4093–4099, 2020.
- [19] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, PMLR, 2016.
- [20] J. Harrison, A. Sharma, and M. Pavone, "Meta-learning priors for efficient online bayesian regression," in *Algorithmic Foundations of Robotics XIII: Proceedings of the 13th Workshop on the Algorithmic Foundations of Robotics 13*, pp. 318–337, Springer, 2020.
- [21] T. D. Bui, C. Nguyen, and R. E. Turner, "Streaming sparse gaussian process approximations," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [22] V. Dhiman, M. J. Khojasteh, M. Franceschetti, and N. Atanasov, "Control barriers in bayesian learning of system dynamics," *IEEE Transactions on Automatic Control*, vol. 68, no. 1, pp. 214–229, 2023.
- [23] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Pointwise feasibility of gaussian process-based safety-critical control under model uncertainty," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6762–6769, 2021.
- [24] 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.
- [25] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial intelligence and statistics*, pp. 567–574, PMLR, 2009.
- [26] F. Castañeda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics," in *2021 American Control Conference (ACC)*, pp. 3683–3690, IEEE, 2021.
- [27] V. Gómez-Verdejo and M. Martínez-Ramón, "Adaptive sparse gaussian process," *arXiv preprint arXiv:2302.10325*, 2023.
- [28] L. Csató and M. Opper, "Sparse on-line gaussian processes," *Neural computation*, vol. 14, no. 3, pp. 641–668, 2002.
- [29] Y. He and Y. Zhao, "Adaptive robust control of uncertain euler-lagrange systems using gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, 2022, DOI: 10.1109/TNNLS.2022.3222405.
- [30] H. Wendland, *Scattered data approximation*, vol. 17. Cambridge university press, 2004.
- [31] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE transactions on information theory*, vol. 58, no. 5, pp. 3250–3265, 2012.
- [32] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Conference on Robot Learning*, pp. 654–670, PMLR, 2021.
- [33] J. Buch, S.-C. Liao, and P. Seiler, "Robust control barrier functions with sector-bounded uncertainties," *IEEE Control Systems Letters*, vol. 6, pp. 1994–1999, 2021.
- [34] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman, "Pyro: Deep universal probabilistic programming," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 973–978, 2019.
- [35] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," 2023. Accessed: August 10, 2023.