Received: 5 September 2022 | Revised: 7 December 2022 | Accepted: 22 December 2022

DOI: 10.1002/rnc.6585

RESEARCH ARTICLE

WILEY

Safety-critical control for robotic systems with uncertain model via control barrier function

Sihua Zhang[1] | **Di-Hua Zhai**[1,2] | **Yuhan Xiong**[1] | **Juncheng Lin**[1] | **Yuanqing Xia**[1]

[1]School of Automation, Beijing Institute of Technology, Beijing, People's Republic of China

[2]Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing, People's Republic of China

Correspondence
Di-Hua Zhai, School of Automation, Beijing Institute of Technology, Beijing, People's Republic of China.
Email: zhaidih@bit.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 62173035, 61803033, 61836001

Abstract

Usually, it is difficult to build accurate dynamic models for real robots, which makes safety-critical control a challenge. In this regard, this article proposes a double-level framework to design safety-critical controller for robotic systems with uncertain dynamics. The high level planner plans a safe trajectory for low level tracker based on the control barrier function (CBF). First, the high level planning is done independently of the dynamic model by quadratic programs subject to CBF constraint. Afterward, a novel method is proposed to learn the uncertainty of drift term and input gain in nonlinear affine-control system by a data-driven Gaussian process (GP) approach, in which the learning result of uncertainty in input gain is associated with CBF. Then, a Gaussian processes-based control barrier function (GP-CBF) is designed to guarantee the tracking safety with a lower bound on the probability for the low level tracker. Finally, the effectiveness of the proposed framework is verified by the numerical simulation of UR3 robot.

KEYWORDS

control barrier function, Gaussian processes, robotic system, safety-critical control, uncertainty

1 | INTRODUCTION

Robots in real-world applications often work in complex environments, for example, the collaborative robots in industrial scenarios[1,2] and aerial robots in reconnaissance.[3] For the successful applications of these robots in different scenarios, safety is one of the most fundamental prerequisites.[4] To guarantee the safety of dynamic systems, several approaches have been investigated. A common technique for robotic manipulators to achieve real-time obstacle avoidance is the artificial potential field method.[5] Optimization-based methods with safety constraints are also used for designing safety-critical controller, such as model predictive control (MPC).[6] In addition, the control barrier function (CBF) proposed in Reference 7 has also been widely applied to ensure the safety of dynamic system in the optimal control framework. Many CBF works have been developed for systems with accurate models, low relative degree and without disturbances, see References 8-10. Considering that some robotic systems are high-relative-degree, such as bipedal and car-like robots, the exponential control barrier function (ECBF) and high order control barrier function (HOCBF) are developed,[11,12] which still only work with known dynamics. However, in complex environments, the dynamic models of robots are not exactly known due to parametric uncertainty and disturbances.[13] The presence of model uncertainty will not only degrade the control performance, but also lead to the inability to guarantee the safety of system.[14,15] It is a challenge to design a safety-critical controller for robots in the presence of model uncertainty.

In the presence of model uncertainty, the dynamic model of the systems can be seen as a combination of an estimating component and an uncertain part. The uncertain part quantifies the difference between the real system and the estimating component. When the uncertain part is caused by disturbance, References 16 and 17 applies robust CBF to guarantee the safety of system, where the disturbance is bounded to ϵ-ball in some norm form. For the system with structured parametric uncertainty in model, adaptive control strategy is used.[18] Meanwhile, adaptive control is also combined with CBF to guarantee safety. In Reference 19, adaptive CBF (aCBF) is developed to ensure the forward invariance of a safe set for systems with structured parametric uncertainty through parameter adaptation. Due to the conservatism caused by the strongly restrictive condition of aCBF, robust adaptive CBF (RaCBF) is developed in Reference 20. It incorporates set membership identification (SMID)[21] to estimate the model uncertainty. Although RaCBFs is less restrictive than aCBFs, RaCBFs can still be conservative if the maximum parameter error is large. Furthermore, aCBFs and RaCBFs can be applied to system where the structured parametric uncertainty only affects the drift term of dynamic model. Reference 22 proposes an adaptive quadratic program-based control Lyapunov-barrier function (QP-CLBF) approach for nonlinear systems with uncertain drift term and unknown control coefficient, in which a filtering-based concurrent learning (FCL) adaptive technique is used to guarantee simultaneous exponential convergence of uncertain parameters in drift term and control coefficient. However, it requires that the unknown model can be linearly parameterized.

As a way to deal with model uncertainty, learning-based methods have also been developed in recent years.[23,24] In Reference 25, extreme learning machines technique is used to learn the drift term of dynamic model and barrier certificate to ensure safety. For safety-critical control of systems with additive stochastic disturbances, Reference 26 develops an adaptive control framework, namely the leveraging stochastic control Lyapunov function (CLF) and stochastic CBF[27] along with tractable Bayesian model learning, in which the performances of Gaussian process (GP) regression,[28] dropout neural networks[29] and ALPaCA[30] are compared. However, the above methods only consider the uncertainty of drift term in nonlinear affine-control system. Reference 31 models the uncertain parts of drift term and input gain as a union of convex hulls that are learned via GP, which is utilized in quadratic program-based control barrier function (QP-CBF) for long-term autonomy applications. However, the estimation method for uncertainty in input gain is not universal. Reference 32 employs a matrix variate Gaussian process regression to learn both the drift term and the input gain terms of nonlinear control-affine system at the same time, where the probabilistic CBF-based self-triggered controller obtained by a deterministic second order cone program (SOCP) is used to ensure safety of systems. Although this method can estimate the unknown dynamics for systems with arbitrary relative degree, the sample is complicated for high-relative-degree systems. Moreover, the probabilistic safety constraints can only be used in conjunction with ECBF for high-relative-degree systems at present.

In this article, a double-level safety-critical control framework is proposed for the robotic system with uncertain model. First, the high level planner plans a safe trajectory based on CBF, which is used as the reference input of low level tracker. Second, in low level tracker, the uncertainty parts of drift term and input gain in dynamic model are estimated by GP. After getting the estimations, the CBF is generalized as Gaussian process-based CBF (GP-CBF) to guarantee the safety constraint with a lower bound on the probability. Finally, the effectiveness of proposed method is verified via the obstacle avoidance simulation of UR3 robot.

The main contributions of this article are summarized as follows.

1. The proposed framework combines the high level planning and low level tracking to achieve safety-critical control. To guarantee the constraints satisfied for the double-level framework, four sufficient conditions are investigated.
2. A novel GP method is proposed to learn the uncertain parts of drift term and input gain in affine control system.
3. Gaussian process-based control barrier function (GP-CBF) is proposed for the system with uncertain model to ensure safety with high probability. Meanwhile, it can be easily generalized to Gaussian process-based high order control barrier function (GP-HOCBF) for high-relative-degree systems.
4. The control-synthesis procedure using CBF and the proposed GP-CBF is specialized to robots for simulated application.

The article is structured as follows. Section 2 revisits some preliminaries about CBF, HOCBF and the optimal problem with CBF for dynamic system. Section 3 presents the problem statement. Section 4 provides the control framework, learning method for model uncertainty and the concept of GP-CBF. Section 5 introduces the controller synthesis for robotic system. The effectiveness of the proposed control method is demonstrated by simulations in Section 6. Finally, conclusions are given in Section 7.

Notations: $\mathbb{R}^n$ denotes the n-dimensional Euclidean space and $\mathbb{R}^{n\times n}$ denotes a space of real matrices with n rows and m columns. $\mathcal{N}(\mu,\omega)$ denotes the multivariate normal distribution with mean vector $\mu \in \mathbb{R}^n$ and covariance matrix $\omega \in \mathbb{R}^{n\times n}$. $\|\cdot\|$ denotes the 2-norm of the vector. $|\cdot|$ denotes the absolute value. $L_f b(x)$ denotes the Lie derivation of $b(x)$ along f at x, and $L_f b(x) = \frac{\partial b(x)}{\partial x} f(x)$. $L_f^m b(x)$ denotes the m-order Lie derivation of $b(x)$ along f at x. q, $\dot{q}$, and $\ddot{q}$ denote the joint position, velocity and acceleration of robotic system, respectively. f_0 and g_0 denote the drift term and input gain of nominal model of robotic system respectively. D_a and D_m denote the uncertain part of drift term and input gain respectively. μ_d^a and σ_d^a denote the mean and standard-deviation predictions for the uncertain part of drift term respectively. μ_d^δ and σ_d^δ denote the mean and standard-deviation predictions for the uncertain part $\Delta(x,u)$ respectively. Q and R denote the prediction boundaries of $\frac{\partial b(x)}{\partial x} D_m$.

2 | PRELIMINARIES

2.1 | Control barrier function

Consider an affine control system

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

where $x \in \mathbb{R}^n$, the drift term $f : \mathbb{R}^n \to \mathbb{R}^n$ and the input gain $g : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ are locally Lipschitz continuous functions. For any initial state $x(t_0) \in \mathbb{R}^n$, $x(t)$ is the unique solution to system (1) on a maximum time interval $I(x_0) = [t_0, \tau_{\max})$. $u \in U \subset \mathbb{R}^n$ is the control input. U denotes the input constraint set and satisfies

$$U = \{u \in \mathbb{R}^n : u_{\min} \le u \le u_{\max}\}. \tag{2}$$

Given a closed set C defined by a continuous differentiable function $b(x) : \mathbb{R}^n \to \mathbb{R}$ as

$$C = \{x \in \mathbb{R}^n : b(x) \ge 0\}, \tag{3}$$

$$\mathrm{Int}(C) = \{x \in \mathbb{R}^n : b(x) > 0\}, \tag{4}$$

$$\partial C = \{x \in \mathbb{R}^n : b(x) = 0\}, \tag{5}$$

It is assumed that C is nonempty and has no isolated point. If for every $x_0 \in C$, the state $x(t)$ always stays in the set C for $t \in I(x_0)$, the set C is called forward invariant.[33] Then the safety of system (1) is guaranteed and the set C is called safe set.

To ensure set invariance, the CBF is derived. First, some important definitions are introduced.

Definition 1 (Class $\mathcal{K}$ function[34]). A continuous function $\alpha : [0, a) \to [0, \infty)$, $a > 0$, is a class $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$.

Definition 2 (Relative degree[34]). For a continuous differentiable function $b(x) : \mathbb{R}^n \to \mathbb{R}$ with respect to system (1), the relative degree is the number of times it needs to be differentiated along with its dynamics until the control input explicitly shows in the corresponding derivative.

Suppose that the relative degree of function $b(x)$ is m, and the inequality $b(x) \ge 0$ is used as a constraint with the relative degree of m. If $m = 1$, the definition of CBF is given.

Definition 3 (Control barrier function). Given a set C as in (3), $b(x)$ is a CBF for system (1) if there exists a class $\mathcal{K}$ function $\alpha(\cdot)$ such that

$$\sup_{u\in U} [L_f b(x) + L_g b(x)u + \alpha(b(x))] \ge 0, \forall x \in C. \tag{6}$$

If $b(x)$ is a CBF, the admission set of control input is defined as

$$K_{\mathrm{cbf}}(x) = \{u \in U : L_f b(x) + L_g b(x)u + \alpha(b(x)) \ge 0\}. \tag{7}$$

The following lemma guarantees the set C is forward invariant.

Lemma 1. *Given the set C is defined by (3) for a continuous differentiable function $b(x)$, if $b(x)$ is a CBF, then Lipschitz continuous control input $u(t) \in K_{\text{cbf}}(x)$ renders the set C forward invariant.*

2.2 | High order control barrier function

If the relative degree of $b(x)$ satisfies $m > 1$, the CBF can not be used to guarantee the forward invariance of set since the control input u is no longer exhibited in (6). Therefore, the HOCBF is proposed. A sequence of functions $\psi_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{0, \dots, m\}$ is first defined as

$$\begin{aligned} &\psi_0(x) = b(x), \\ &\psi_i(x) = \dot{\psi}_{i-1}(x) + \alpha_i(\psi_{i-1}(x)), \quad i \in \{1, \dots, m\}, \end{aligned} \tag{8}$$

where $\alpha_i(\cdot)$ denotes $(m-i)$th order differentiable class $\mathcal{K}$ function. A sequence of sets $C_i, i \in \{1, \dots, m\}$ are then defined in the form of

$$C_i = \{x \in \mathbb{R}^n : \psi_{i-1}(x) \geq 0\}, \quad i \in \{1, \dots, m\}. \tag{9}$$

Given the functions $\psi_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}, i \in \{0, \dots, m\}$, the definition of HOCBF is as below.

Definition 4 (HOCBF[12]). A function $b(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a HOCBF of relative degree m for system (1) if there exist $(m-i)$th order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, \dots, m-1\}$, and a class $\mathcal{K}$ function α_m such that

$$\sup_{u \in U} [L_f^m b(x) + L_g L_f^{m-1} b(x) u + S(b(x)) + \alpha_m(\psi_{m-1}(x))] \geq 0, \tag{10}$$

for all $x \in C_1 \cap \dots \cap C_m$. The Equation (10) equals to $\psi_m(x) \geq 0$. $S(b_i(x))$ denotes the remaining Lie derivative along f with degree less than or equal to $m-1$, that is,

$$S(b_i(x)) = \sum_{i=1}^{m-1} L_f^i (\alpha_{m-i} \circ \psi_{m-i-1})(x). \tag{11}$$

Similar to Lemma 1, the following result also guarantees the forward invariance of set C.

Lemma 2 (12). *The set $C_1 \cap \dots \cap C_m$ is forward invariant for system (1) if $x(0) \in C_1 \cap \dots \cap C_m$ and $b(x)$ is a HOCBF. The Lipschitz continuous control input u belongs to the set*

$$K_s(x) = \left\{ u \in U : L_f^m b(x) + L_g L_f^{m-1} b(x) u + S(b(x)) + \alpha_m(\psi_{m-1}(x)) \geq 0 \right\}. \tag{12}$$

2.3 | Optimal control with CBF

To find a control policy for system (1), the optimal control problem is considered by combining CBF constraint with quadratic costs in control u.

$$\begin{aligned} J(u(t)) &= \int_0^T \|u(t) - u_{\text{nom}}(t)\|^2 dt \\ s.t. \quad & L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0 \\ & u_{\min} \leq u \leq u_{\max}, \end{aligned} \tag{13}$$

where u_{nom} is a nominal feedback controller. When $u_{\text{nom}} \notin K_{\text{cbf}}(x)$, the CBF constraint will minimally modify the nominal controller to ensure safety. Replacing CBF condition with HOCBF condition can handle a constraint of high relative degree.

The optimal control problem is usually solved point-wise, where the time interval $[0, T]$ is divided into a finite number of intervals $[t_k, t_{k+1})$, $k = 0, 1, 2 \dots n$. Besides, the constraint is linear in control and the states are fixed at each interval,

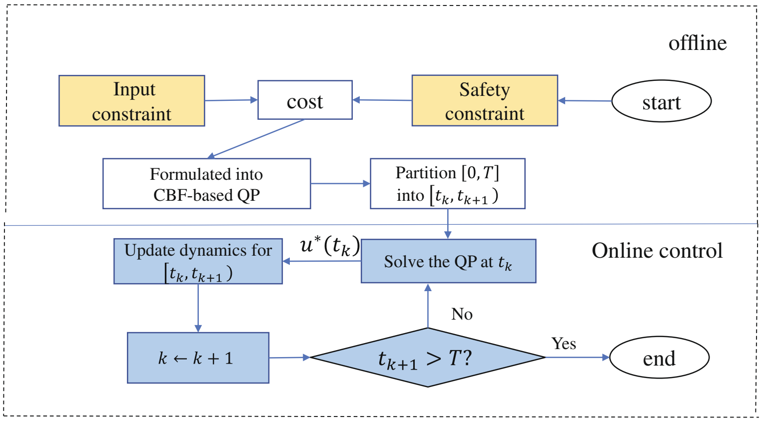


FIGURE 1 Flowchart for solving a CBF constrained optimal control problem.

so that the optimization problem eventually becomes a quadratic programming (QP) as

$$\begin{aligned} & u^* = \arg\min_{u} \|u - u_{\text{nom}}\|^2 \\ s.t. \quad & L_f b(x) + L_g b(x) u + \alpha(b(x)) \geq 0, \\ & u_{\min} \leq u \leq u_{\max}, \end{aligned} \tag{14}$$

Then, the optimal control input is obtained by solving a QP at each interval, and updating states through dynamics (1). The whole process of solving the optimal control problem with CBF constraint to guarantee the safety is shown in Figure 1.

Remark 1. Most existing work on implementing safety-critical control with CBF solves the QP (14) in each interval using the above method.[4,12] This conversion is first applied in Reference 35, which proves the control input from QP is locally Lipschitz continuous and a closed-form expression can be given for control input. It is important to note that this method is computationally efficient, but continuous safety may not be satisfied between time intervals. For this problem, References 36 and 37 present improved methods by bounding the time derivative of the CBF between time intervals. Given that the above solution is complicated and is out of the main focus of this article, the proposed framework only sets the time intervals to be small to avoid this problem.

Remark 2. Noted that this method works conditioned on the fact that the QP (14) at every time interval is feasible. Some methods can be used to guarantee the feasibility of QP (14). The penalty method is proposed in Reference 12 to guarantee the feasibility of QP by adding penalty coefficient in class $\mathcal{K}$ function. Reference 38 introduces adaptive CBF to resolve the conflict between CBF constraint and input constraint by introducing penalty functions in the definition of CBF and defining auxiliary dynamics for these penalty functions. Reference 39 provides a method to find sufficient conditions to guarantee the feasibility of QP subject to CBF constraint and input constraint. Based on the above methods, the feasibility of QP (14) can be guaranteed. Noted that, the above methods are easily implemented in QP (14). In this case, given that the feasibility of QP is not the main focus of this article, we will not analyze the feasibility of this part in depth to present our proposed control framework as concisely as possible.

As the dynamic model (1) appears in the CBF constraint to enforce safety, this optimal control method works under the condition that the dynamics model is accurate. However, this is not guaranteed for practical robotics system. So this article shows how to ensure safety under uncertain dynamic models.

3 | PROBLEM FORMULATION

This section provides assumptions on the model uncertainty of robotic systems, and formulates the safety-related dynamics learning problem.

Consider the Euler–Lagrangian dynamics of a robotic system in the form of

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u, \tag{15}$$

where $q \in \mathbb{R}^n$, $M(q) \in \mathbb{R}^{n\times n}$, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$, and $G(q) \in \mathbb{R}^n$ are the inertia matrix, Coriolis-centrifugal matrix and gravitational term respectively. $M(q)$ is symmetric and positive-definite.[40] $u = [\tau_1, \tau_2, \dots, \tau_n]^T \in U$ is the control torque.

Due to the existence of model uncertainty caused by parametric error and disturbances, the matrices $M(q)$, $C(q,\dot{q})$ and $G(q)$ are not precisely known. So, control synthesis is done by estimating those unknown matrices to approximate the true dynamics.

$$\hat{M}(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + \Delta_d(q,\dot{q},\ddot{q}) = u, \tag{16}$$

where $\hat{M}(q)$, $\hat{C}(q,\dot{q})$, and $\hat{G}(q)$ represent the nominal parts of the matrices $M(q)$, $C(q,\dot{q})$, and $G(q)$, respectively. $\Delta_d(q,\dot{q},\ddot{q})$ is total uncertainties. The uncertain dynamics will manifest in the CBF constraint, which makes it impossible to verify that a given control input satisfies the inequality in (6) and lead to unsafe behavior.

In order to formulate the problem, some assumptions on the uncertainties are given.

Assumption 1 (41). The uncertain dynamic $\Delta_d(q,\dot{q},\ddot{q})$ has low complexity under the reproducing kernel Hilbert space (RKHS) norm, and it has bounded RKHS norm with respect to know kernel k, that is, $\|\Delta_{d_i}\|_k < \infty$, for all $i = 1, 2, \dots n$.

Assumption 2. The measurements q and $\dot{q}$ are accessed, and the uncertainty Δ_d with additive noise $\omega \backsim \mathcal{N}(0, \delta_\omega)$ is obtained through measurements q, $\dot{q}$, and u.

In addition, the nominal dynamic model in (16) satisfies the following assumption.

Assumption 3. The relative degrees of CBF for the nominal dynamic model and uncertainty part in (16) are the same as that of the true system (15). And the matrix $\hat{M}(q)$ is symmetric positive-definite.

Remark 3. Intuitively, it is required that a set in the state space to be safe for the true model of the system as well as for the uncertain model, which is equivalent to Assumption 3.[42] Proposition 2 in Reference 43 shows that Assumption 3 is tenable.

Based on the above assumptions, the main safety-critical control synthesis problem in this article is formally defined next.

Consider an affine system with uncertain part as

$$\dot{x} = f_0(x) + g_0(x)u + \Delta(x,u), \tag{17}$$

where $x \in \mathbb{R}^n$, $f_0 : \mathbb{R}^n \to \mathbb{R}^n$, and $g_0 : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ are locally Lipschitz and known, $\Delta(x,u)$ is uncertain part.

For the robotic system (16) satisfying Assumptions 1 and 3, the dynamic model can be formulated as the form of (17), in which $x = [q,\dot{q}]^T$, $f_0(x) = [\dot{q}, -\hat{M}(q)^{-1}(\hat{C}(q,\dot{q})\dot{q} + \hat{G}(q))]^T$, $g_0(x) = [0, \hat{M}(q)^{-1}]^T$. The form of uncertain part in (16) is

$$\Delta_d(q,\dot{q},\ddot{q}) = (M(q) - \hat{M}(q))\ddot{q} + (C(q,\dot{q}) - \hat{C}(q,\dot{q}))\dot{q} + (G(q) - \hat{G}(q)), \tag{18}$$

where the acceleration $\ddot{q}$ depends on the control input u as shown in (15). Therefore, the uncertain part in (16) is transformed as the form of $\Delta(x,u)$ via (15), which is derived as

$$\Delta(x,u) = D_a(x) + D_m(x)u, \tag{19}$$

where $D_a(x) = [0, -M(q)^{-1}(C(q,\dot{q})\dot{q} + G(q)) + \hat{M}(q)^{-1}(\hat{C}(q,\dot{q})\dot{q} + \hat{G}(q))]^T$ is the uncertain part of drift term, $D_m(x) = [0, M(q)^{-1} - \hat{M}(q)^{-1}]^T$ is the uncertain part of input gain.

The aim is to find a control policy to make sure that the robotic system can track trajectories while being subjected to certain safety constraints. The safety of robotic system is described by the superlevel set of a continuously differentiable function $b(q) : \mathbb{R}^n \to \mathbb{R}$ as

$$S = \{q \in \mathbb{R}^n : b(q) \geq 0\}. \tag{20}$$

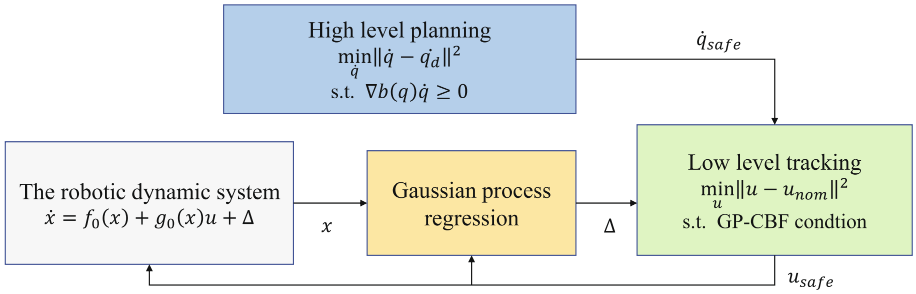


FIGURE 2 Double-level safety-critical control framework.

Due to the uncertain dynamics, it is often difficult to find an solution to above problem. Therefore, this article provides a double-level framework to solve this problem, which will be introduced in the next section.

4 | DOUBLE-LEVEL SAFETY-CRITICAL FRAMEWORK

In this section, four properties associated with the high level trajectory planner and low level safety tracker are investigated. Gaussian process is used to model the unknown dynamics and a notion of GP-CBF is also developed, which depends on the learning models to guarantee safety.

4.1 | Double-level framework and its associated properties

First, a double-level framework is proposed as shown in Figure 2, in which high level design plans a safe trajectory planner to get the safe joint angel q_{safe} as the reference input of low level controller. The low level tracker is a tracking controller to track the safe trajectory in the presence of model uncertainty. For the high level planner, let the desired trajectory of robotic end-effector be $P(t) \in \mathbb{R}^n$, then the tracking error is defined as

$$e = T(q) - P(t), \tag{21}$$

$$\dot{e} = J_y(q)\dot{q} - \dot{P}(t), \tag{22}$$

where $T(q) \in \mathbb{R}^n$ is the position of end-effector, $J_y(q)$ is the Jacobian matrix. High level planner needs to satisfy the following properties to plan a safe trajectory via $\dot{q}$.

Property 1 (High level safety). The high level planner adds a constraint on $\dot{q}$ to guarantee the safety, that is, if the initial position $q(0) \in S$, then the constrained $\dot{q}$ makes joint angle $q(t) \in S$ for all $t > 0$.

Property 2 (High level tracking). The high level planner gets the safe joint angle $q_{\text{safe}}(t) \in S$ on the basis of ensuring safety and also makes the tracking error e converge to 0.

The low level tracker guarantees that the planned trajectory can be safely tracked by the true system through controller u. And the following properties are required.

Property 3 (Low level safety). The low level tracker adds a constraint on control input u to guarantee the control safety, that is, if the initial position $q(0) \in S$, then the control u makes joint angle $q(t) \in S$ for all $t > 0$.

Property 4 (Low level tracking). The low level tracker gets the control input u on the basis of ensuring safety and also makes the tracking error $q(t) - q_{\text{safe}}(t) \to 0$.

4.2 | Gaussian process regression

Since the low level tracker depends on the dynamic model, the unmodeled dynamics $\Delta(x, u)$ may lead to unsafe. To solve this problem, Gaussian process (GP) is used for estimating $D_a(x)$ and $D_m(x)$. GP is a kernel-based regression method for estimating the functions from data. Compared to other regression methods, GP is data efficient, which can obtain an approximate estimation of the unmodeled dynamics using a small number of data-points. GP is dentally introduced in the following.

Given training data $\mathcal{D} = \{z^{(i)}, r^{(i)}\}_{i=1}^N$, where $z^{(i)}$ is the ith input data, and $r^{(i)}$ is the ith output data with measurement noise as shown in Assumption 2. The objective of GP is to describe the model of the uncertain dynamics as $r \backsim \mathcal{N}(\mu_d(z), \sigma_d^2(z))$, where $\mu_d(z)$ is the mean and $\sigma_d^2(z)$ is the covariance. The mean and covariance are parameterized by a kernel function $k(z^{(i)}, z^{(j)})$, which describes the similarity between two data points $z^{(i)}, z^{(j)}$. Based on the training data and Assumption 2, the posterior distribution of unmodeled dynamics at a query point z^* by mean and covariance as

$$\mu_d(z^*) = \overline{k}^T(z^*)(K + \delta_\omega)^{-1}g, \tag{23}$$

$$\sigma_d^2(z^*) = k(z^*, z^*) - \overline{k}^T(z^*)(K + \delta_\omega)^{-1}\overline{k}(z^*), \tag{24}$$

where $\overline{k} = [k(z^{(1)}, z^*), k(z^{(2)}, z^*), \dots, k(z^{(N)}, z^*)]^T \in \mathbb{R}^N$ and the kernel matrix $K \in \mathbb{R}^{N\times N}$ is defined as

$$K = \begin{bmatrix} k(z^{(1)}, z^{(1)}) & \cdots & k(z^{(1)}, z^{(N)}) \\ \vdots & \ddots & \vdots \\ k(z^{(N)}, z^{(1)}) & \cdots & k(z^{(N)}, z^{(N)}) \end{bmatrix}.$$

On the basis of Assumption 1, the difference between the unmodeled dynamics $\Delta(z^*)$ and the mean $\mu_d(z^*)$ can be bounded on the function

$$\mu_d(z^*) - k_\delta \sigma_d(z^*) \leq \Delta(z^*) \leq \mu_d(z^*) + k_\delta \sigma_d(z^*), \tag{25}$$

by probability $(1 - \delta)$, where k_δ is a design parameter that determines δ.

Remark 4. The training of GP becomes problematic as the amount of data increases, because the complexity of computing the matrix inverse in (23) and (24) is $\mathcal{O}(N^3)$. This problem can be overcome by batch training the GP model with random sample data points.

4.3 | Gaussian process-based control barrier function

According to the definition of CBF, the CBF condition for safety-critical control of system (17) is the form as

$$\frac{\partial b(x)}{\partial x}(f_0(x) + g_0(x)u + D_a(x) + D_m(x)u) + \alpha(b(x)) \geq 0, \tag{26}$$

where $D_a(x)$ and $D_m(x)$ need to be estimated by GP.

The training data $\mathcal{D} = \{z^{(i)}, r^{(i)}\}_{i=1}^N$ for $D_a(x)$ is sampled with input signal $u = 0$, in which $z^{(i)} = [q, \dot{q}] \in \mathbb{R}^{2\times n}$ is the input data and $r^{(i)} = D_a(z^{(i)}) + \omega \in \mathbb{R}^n$ is the output data. The estimation of uncertain part in drift term for a query point x^* is

$$\mu_d^a(x^*) - k_\delta \sigma_d^a(x^*) \leq D_a(x^*) \leq \mu_d^a(x^*) + k_\delta \sigma_d^a(x^*), \tag{27}$$

where $\mu_d^a(x^*)$ and $\sigma_d^a(x^*)$ are the mean and standard-deviation predictions for query point x^*.

Note that in contrast to $D_a(x)$, $D_m(x)$ is a matrix in $\mathbb{R}^{n\times n}$. It poses a difficulty in building the dataset $\mathcal{D}$ since the output data $r^{(i)} = D_m(z^{(i)})$ cannot be obtained directly. To this end, an indirect method is proposed to estimate $\frac{\partial b(x)}{\partial x} D_m(x) \in \mathbb{R}^n$.

First, the uncertain part $\Delta(x, u)$ is estimated by GP. The training data $\mathcal{D} = \{z^{(i)}, r^{(i)}\}_{i=1}^N$ for $\Delta(x, u)$ is sampled with nominal controller, in which $z^{(i)} = [x, u] \in \mathbb{R}^{2n}$ is the input data and $r^{(i)} = \Delta(z^{(i)}) + \omega \in \mathbb{R}^n$ is the output data. The estimation of $\Delta(x, u)$ for a query point $z^* = [x^*, u^*]$ is

$$\mu_d^\delta(z^*) - k_\delta \sigma_d^\delta(z^*) \leq \Delta(z^*) \leq \mu_d^\delta(z^*) + k_\delta \sigma_d^\delta(z^*), \tag{28}$$

where $\mu_d^\delta(z^*)$ and $\sigma_d^\delta(z^*)$ are the mean and standard-deviation predictions for query point z^*.

According to (27) and (28), $D_m(x^*)u^*$ can be bounded as

$$\mu_d^\delta(z^*) - \mu_d^a(x^*) - k_\delta\left[\sigma_d^\delta(z^*) + \sigma_d^a(x^*)\right] \leq D_m(x^*)u^* \leq \mu_d^\delta(z^*) - \mu_d^a(x^*) + k_\delta\left[\sigma_d^\delta(z^*) + \sigma_d^a(x^*)\right] \tag{29}$$

with probability $1-\delta$.

Then, $\frac{\partial b(x)}{\partial x}D_m(x^*)u^*$ can be bounded as

$$\begin{aligned}\frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z^*) - \mu_d^a(x^*)\right) - k_\delta\left|\frac{\partial b(x)}{\partial x}\right|[\sigma_d^\delta(z^*) + \sigma_d^a(x^*)] \leq \frac{\partial b(x)}{\partial x}D_m(x^*)u^* \\ \leq \frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z^*) - \mu_d^a(x^*)\right) + k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\left[\sigma_d^\delta(z^*) + \sigma_d^a(x^*)\right].\end{aligned} \tag{30}$$

If there exist query points $z_j^* = [x^*, u_j^*]$, where $j = 1, \ldots, n$, $u_j^* = [0, \ldots, c_j, \ldots, 0]$ and $c_j \neq 0$, the estimation of $\frac{\partial b(x)}{\partial x}D_m(x^*)$ can be bounded as

$$Q \leq \frac{\partial b(x)}{\partial x}D_m(x^*) \leq R, \tag{31}$$

where $Q, R \in \mathbb{R}^n$. For $c_j > 0$, the jth component of Q and R are

$$Q_j = \frac{\frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z_j^*) - \mu_d^a(x^*)\right) - k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\left[\sigma_d^\delta(z_j^*) + \sigma_d^a(x^*)\right]}{c_j},$$
$$R_j = \frac{\frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z_j^*) - \mu_d^a(x^*)\right) + k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\left[\sigma_d^\delta(z_j^*) + \sigma_d^a(x^*)\right]}{c_j}.$$

For $c_j < 0$, the jth component of Q and R are

$$Q_j = \frac{\frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z_j^*) - \mu_d^a(x^*)\right) + k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\left[\sigma_d^\delta(z_j^*) + \sigma_d^a(x^*)\right]}{c_j},$$
$$R_j = \frac{\frac{\partial b(x)}{\partial x}\left(\mu_d^\delta(z_j^*) - \mu_d^a(x^*)\right) - k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\left[\sigma_d^\delta(z_j^*) + \sigma_d^a(x^*)\right]}{c_j}.$$

After $D_a(x)$ and $\frac{\partial b(x)}{\partial x}D_m(x)$ are estimated by GP, the GP-based control barrier function (GP-CBF) can be defined as following.

Definition 5. If the given set C is the superlevel set of a continuously differentiable function $b(x)$ as defined in (3), and the uncertain part in system (17) is estimated by (27) and (31), then the function $b(x)$ is a GP-based control barrier function (GP-CBF) if there exists class $\mathcal{K}$ function $\alpha(\cdot)$ such that

$$\sup_{u\in U}\left[L_{f_0}b(x) + L_{g_0}b(x)u + L_{\mu_d^a}b(x) - k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\sigma_d^a(x) + \min(Qu, Ru) + \alpha(b(x))\right] \geq 0, \quad \forall x \in C. \tag{32}$$

An admissible set of control input is obtained by GP-CBF:

$$\begin{aligned}K_{\text{gp-cbf}}(x) = \Big\{u \in U : L_{f_0}b(x) + L_{g_0}b(x)u + L_{\mu_d^a(x)}b(x) \\ - k_\delta\left|\frac{\partial b(x)}{\partial x}\right|\sigma_d^a(x) + \min(Qu, Ru) + \alpha(b(x)) \geq 0\Big\}.\end{aligned} \tag{33}$$

The next result provides a probabilistic guarantee for the forward invariance of set C.

Theorem 1. *If $b(x)$ is a GP-CBF, the Lipschitz continuous control input $u(t) \in K_{\text{gp-cbf}}(x)$ renders set C forward invariant with probability at least $1-\delta$.*

Proof. From (27) and (31), the following probability can be obtained.

$$\mathbb{P} = \left\{ \forall x \in X, \exists u \in U : \frac{\partial b(x)}{\partial x}(f_0(x) + g_0(x)u + \overline{D}_a(x) + \overline{D}_m(x)u) + \alpha(b(x)) \geq 0 \right\} = 1 - \delta, \tag{34}$$

where $\overline{D}_a(x)$ and $\overline{D}_m(x)$ satisfy (27) and (31), respectively.

Define the events $A_1, \dots, A_4$, where A_1 means that there exists u satisfying (32) for all states, A_2 represents that there exists u satisfying

$$\frac{\partial b(x)}{\partial x}(f_0(x) + g_0(x)u + \overline{D}_a(x) + \overline{D}_m(x)u) + \alpha(b(x)) \geq 0, \tag{35}$$

for all states, A_3 represents that there exists u satisfying (6) for all states, and A_4 represents that the set C is forward invariant.

Since the events $A_2 \rightarrow A_3$ and $A_3 \rightarrow A_4$ are mutually independent, the relationship of events is $\mathbb{P}\{A_1 \rightarrow A_4\} \geq \mathbb{P}\{A_2 \rightarrow A_4\} = \mathbb{P}\{A_2 \rightarrow A_3\}\mathbb{P}\{A_3 \rightarrow A_4\}$. The probability $\mathbb{P}\{A_3 \rightarrow A_4\} = 1$ is obtained from Lemma 1, and $\mathbb{P}\{A_2 \rightarrow A_3\}$ is shown in (34). Thus, the GP-CBF $b(x)$ renders set C forward invariant with probability at least $1 - \delta$. ■

5 | CONTROLLER SYNTHESIS

In this section, how to synthesize safe controller satisfying Properties 1–4 for uncertain system by GP will be discussed. First, CBF is used to plan a safe trajectory in high level loop. Then, GP estimates the uncertain part of dynamic model, and a GP-CBF based controller is designed to track the safe trajectory in low lever loop.

For robotic systems, the high level planner plans a safe trajectory, which is determined by safe velocity $\dot{q}_{\text{safe}}$. In order to satisfy Property 1, CBF $b(q)$ is used to ensure the forward invariance of safety set defined in (20), and it adds a constraint on velocity $\dot{q}$ as

$$\frac{\partial b(q)}{\partial q}\dot{q} + \alpha(b(q)) \geq 0. \tag{36}$$

For Property 2, the trajectory errors are described by (21) and (22). If $\dot{q}$ satisfies $J_y(q)\dot{q} = \dot{P} - \gamma e$ for $\gamma > 0$, $\dot{e} = -\gamma e \Rightarrow e(t) = \exp(-\gamma t)e(0)$. Therefore, by picking

$$\dot{q}_d = J_y(q)^{\dagger}(\dot{P} - \gamma e(t)), \tag{37}$$

where $J_y(q)^{\dagger} = J_y(q)^T(J_y(q)J_y(q)^T)^{-1}$, it holds $e \rightarrow 0$, then Property 2 is satisfied.

According to the above analysis, a QP-CBF is formulated unifying (36), (37) and physical limits on velocity to obtain the safe velocity $\dot{q}_{\text{safe}}$.

$$\begin{aligned} \dot{q}_{\text{safe}} &= \arg\min_{\dot{q}} \|\dot{q} - \dot{q}_d\|^2 \\ s.t.\ & \frac{\partial b(q)}{\partial q}\dot{q} + \alpha(b(q)) \geq 0 \\ & \dot{q}_{\min} \leq \dot{q} \leq \dot{q}_{\max}. \end{aligned} \tag{38}$$

The low level tracker satisfying the Properties 3 and 4 is designed to track the safe velocity $\dot{q}_{\text{safe}}$ based on dynamic model. A nominal controller and the GP-CBF condition (32) are designed for Properties 3 and 4, respectively. The schematic of controller-synthesis framework in low level loop is depicted in Figure 3, where a feedback control loop between the GP, GP-CBF and the dynamic model is created.

Remark 5. The nominal controller in low level tracking loop should ensure that the robot can track the safe trajectory. For robotic systems with uncertain dynamic model, many works have been proposed to guarantee the tracking performance,[44,45] among which PID controller is one of the most convenient control methods to implement. Therefore, in this article, a PID controller is designed as the nominal controller.

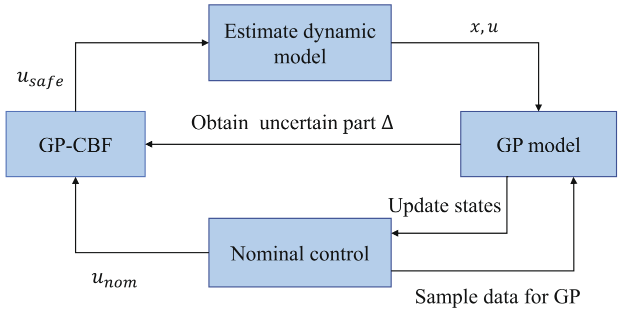
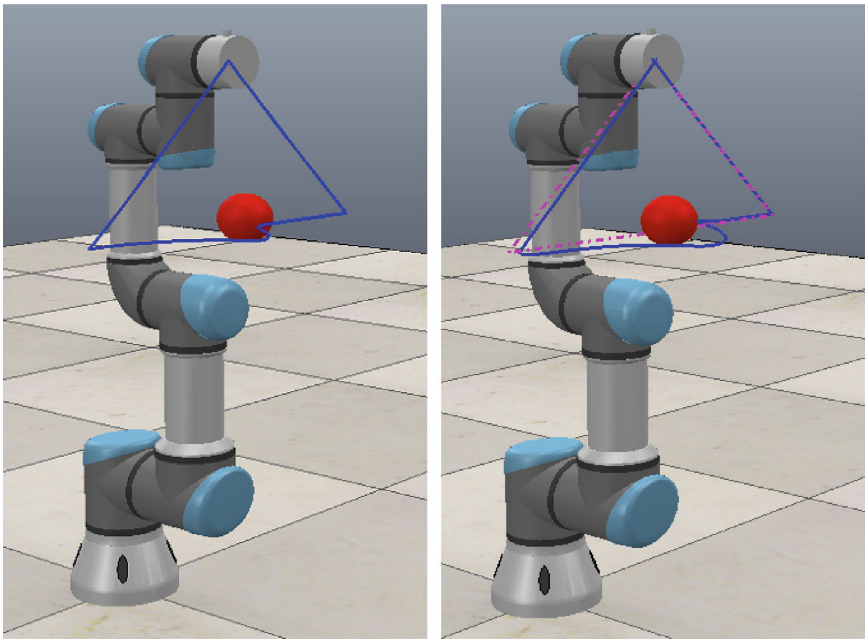


FIGURE 3 Schematic diagram of the proposed controller-synthesis framework using GP-CBF in low level tracker loop.



(A) (B)

FIGURE 4 Trajectories of robot end-effector. (A) The planning safe trajectory. (B) The tracking trajectory with different controllers.

Obviously, the relative degree of $b(q)$ is two for the robotic system. The GP-CBF needs to be expanded to Gaussian process-based HOCBF (GP-HOCBF) to guarantee the safety of robotic system.

For a continuous differentiable function $b(x) : \mathbb{R}^n \to \mathbb{R}$, a sequence of functions $\psi_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{0, \dots, m-1\}$ are first defined as

$$\psi_0(x) = b(x)$$
$$\psi_i(x) = \frac{\partial \psi_{i-1}(x)}{\partial x}\left(f_0(x) + \mu_d^a(x)\right) - k_\delta \left|\frac{\partial \psi_{i-1}(x)}{\partial x}\right| \sigma_d^a(x) + \alpha_i(\psi_{i-1}(x)), \tag{39}$$

where $\alpha_i(\cdot)$ is the $(m-i)$th order differentiable class $\mathcal{K}$ function. A sequence of sets $C_i, i \in \{1, \dots, m\}$ are shown in (9). Then the GP-HOCBF can be defined as:

Definition 6 (GP-HOCBF). A function $b(x) : \mathbb{R}^n \to \mathbb{R}$ is a Gaussian process-based HOCBF (GP-HOCBF) of relative degree m for system (17), if there exist $(m-i)$th order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, \dots, m\}$, and a class $\mathcal{K}$ function α_m such that

$$\frac{\partial \psi_{m-1}(x)}{\partial x}\left(f_0(x) + \mu_d^a(x) + g_0(x)u\right) - k_\delta \left|\frac{\partial \psi_{m-1}(x)}{\partial x}\right| \sigma_d^a(x)$$
$$+ \min(Qu, Ru) + \alpha_m(\psi_{m-1}(x)) \geq 0, \tag{40}$$

for all $x \in C_1 \cap \cdots \cap C_m$.

The controller for tracking the safe velocity is designed through a QP subject to GP-HOCBF as

$$\begin{aligned} u_{\text{safe}} &= \arg\min_{u} \|u - u_{\text{nom}}\|^2 \\ s.t.\ & L_{f_0}^2 b(q) + L_{g_0} L_{f_0} b(q) u + \frac{\partial b(q)}{\partial q}\mu_d^a - k_\delta \left|\frac{\partial b(q)}{\partial q}\right| \sigma_d^a \\ & + \min(Qu, Ru) + (k_1 + k_2) L_{f_0} b(q) + k_1 k_2 b(q) \geq 0, \\ & u_{\min} \leq u \leq u_{\max}, \end{aligned} \tag{41}$$

where $k_1 > 0$ and $k_2 > 0$ are parameters of the class $\mathcal{K}$ functions.

Lastly, as the controllers in high level loop and low level loop are designed respectively, the safety of robotic system with uncertain dynamics is guaranteed. The double-level safety-critical control framework is summarized in Algorithm 1.

Algorithm 1. Double-level safety-critical control framework

Input: The desired trajectory P, CBF $b(q)$, the estimated matrix $\hat{M}(q)$, $\hat{C}(q, \dot{q})$, and $\hat{G}(q)$, the parameters in the QP γ, k_1, k_2, the parameters of nominal controller k_p, k_d, k_i

Output: The training data $\mathcal{D}^a$ and $\mathcal{D}^m$, the safe controller u_{safe}

$\mathcal{D} = \emptyset$

for $t = t_i, i = 1, 2, \dots, N$ **do**

  Solve the QP (38) get $\dot{q}_{\text{safe}}(t)$

  Update the $q_{\text{safe}}(t_i) = q_{\text{safe}}(t_{i-1}) + \dot{q}_{\text{safe}}(t)$

end for

for $t = t_i, i = 1, 2, \dots, N$ **do**

  get $x(t_i) = [q(t_i), \dot{q}(t_i)]$ and $D_a(x(t_i))$ with $u = 0$

  get $x(t_i)$ and $\Delta(x(t_i), u_{\text{nom}}(t_i))$ with $u = u_{\text{nom}}(t_i)$

  $\mathcal{D}_i^a \leftarrow ([x(t_i)], D_a(x(t_i)))$, $\mathcal{D}_i^m \leftarrow ([x(t_i), u_{\text{nom}}(t_i)], \Delta(x(t_i), u_{\text{nom}}(t_i))$

  $\mathcal{D}^a = \mathcal{D}^a \cup \mathcal{D}_i^a$, $\mathcal{D}^m = \mathcal{D}^m \cup \mathcal{D}_i^m$

end for

for $t = t_i, i = 1, 2, \dots, N$ **do**

  train GP models by sampling random from $\mathcal{D}^a$, $\mathcal{D}^m$

  predict the μ_d^a and σ_d^a for $D_a(q(t_i), \dot{q}(t_i))$

  predict the Q and R for $\frac{\partial b(q)}{\partial q} D_m(q(t_i), \dot{q}(t_i))$

  solve the QP (41) get $u_{\text{safe}}(t_i)$;

  update the states to get $q(t_{i+1})$, $\dot{q}(t_{i+1})$

end for

Remark 6. In learning process, a very important property is that the data has to be independently and identically distributed. However, the training data sampled along the safe trajectory with the nominal controller violates this property. Similar to Reference 46, randomizing the samples can solve the problem and reduce the variance of the updates.

6 | SIMULATION

In this section, to verify the effectiveness of the proposed framework, it is simulated on a virtual robot UR3 located in the scene designed in Coppeliasim. The end-effector of UR3 robot is required to achieve an obstacle avoidance task.

The safety constraint is described by a continuously differentiable function $b(q) : \mathbb{R}^6 \to \mathbb{R}$ as:

$$b(q) = (x(q) - x_0)^2 + (y(q) - y_0)^2 + (z(q) - z_0)^2 - r^2 \geq 0, \tag{42}$$

where $x(q)$, $y(q)$, and $z(q)$ are the coordinates of end-effector. x_0, y_0, and z_0 are the coordinates of circular obstacle.

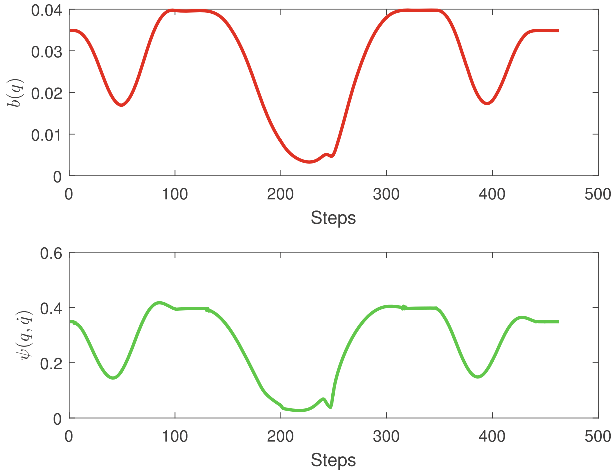


FIGURE 5 Response curves of $b(q)$ and $\psi_1(q,\dot{q})$.

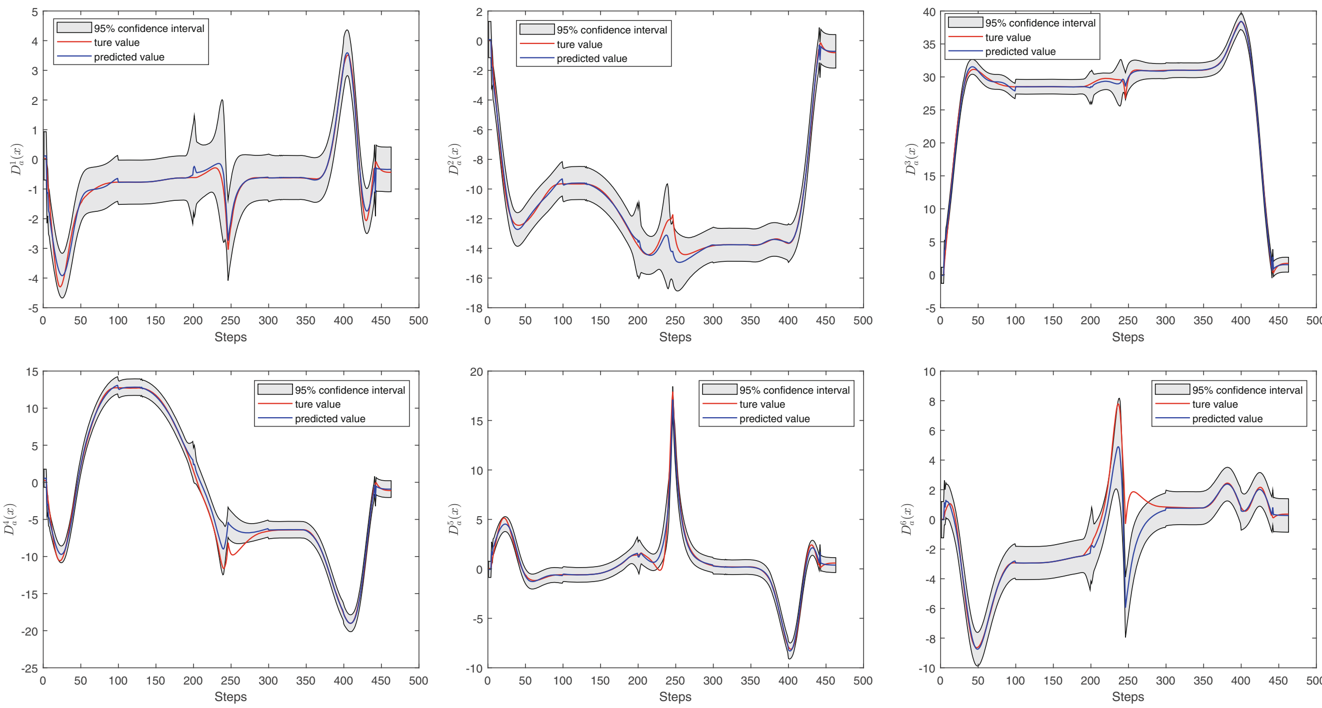


FIGURE 6 Estimates of each dimension of $D_a(x)$ and the 95% confidence interval

The nominal model of UR3 manipulator is set as

$$\begin{cases} \hat{M}(q) = 0.8M(q) \\ \hat{C}(q,\dot{q}) = 0.9C(q,\dot{q}) \\ \hat{G}(q) = 0, \end{cases} \tag{43}$$

where $M(q)$, $C(q,\dot{q})$, and $G(q)$ are given in Reference 40.

Remark 7. Noted that the model uncertainty is greater than 10% of the nominal model, which is a relatively large uncertainty in engineering practice. Therefore, it is convincing to use the above nominal model in simulation to verify the effectiveness of the proposed framework.

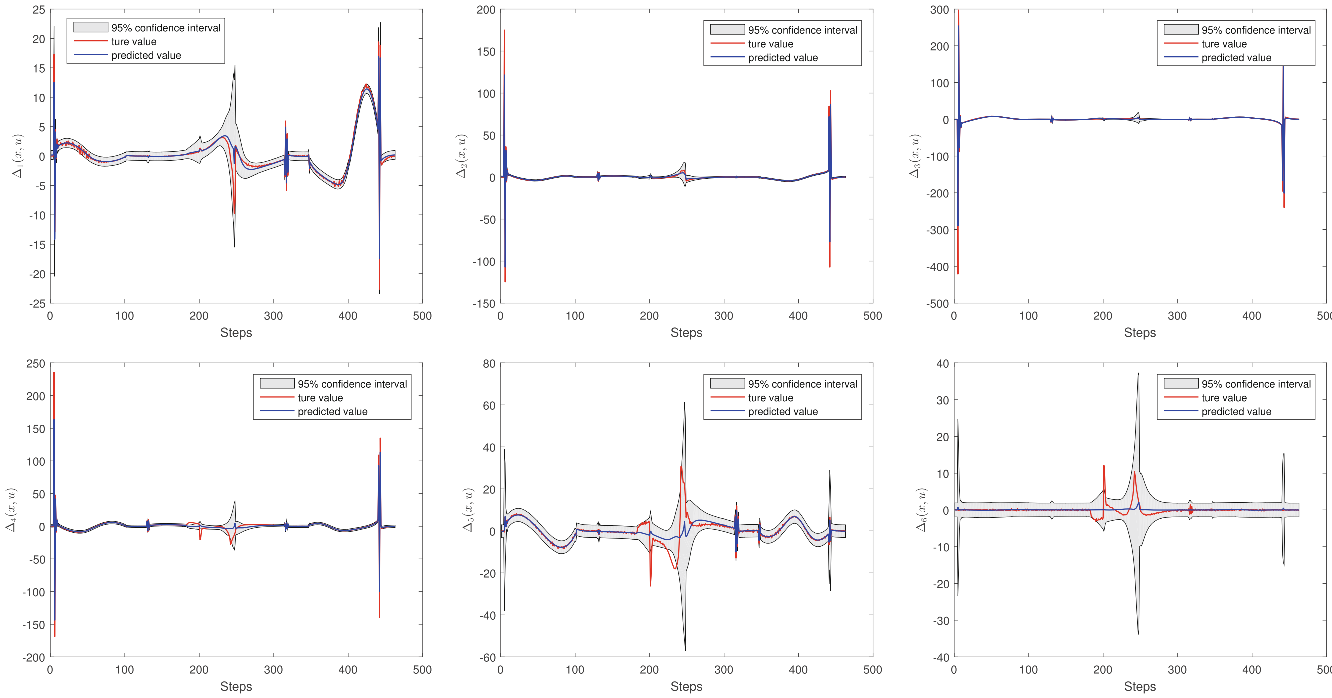


FIGURE 7 Estimates of each dimension of $\Delta(x, u)$ and the 95% confidence interval

For the obstacle-avoiding task, the circular obstacle area is centered at $(-0.2, 0, 0.51)$ with the radius $r = 0.03$. Both the initial and final joint angles are $[0, 0, 0, 0, 0, 0]deg$. The motion range of each joint of the UR3 manipulator is set as $(-\pi, \pi)$. The planning trajectory is completed with the parameter $\gamma = 10$ and class $\mathcal{K}$ function $\alpha(b(q)) = 10b(q)$. Then the uncertain part of dynamic model is learned by GPML toolbox. To track safe trajectory, the class $\mathcal{K}$ functions of GP-CBF set as $\alpha_1(b(q)) = 10b(q)$ and $\alpha_2(\psi_1(q, \dot{q})) = 30\psi_1(q, \dot{q})$.

The planned and tracked safe trajectories are illustrated in Figure 4. In Figure 4B, the tracking trajectory is obtained by using the controller that does not consider the uncertainty of dynamic model (see the pink dotted line). Obviously, it does not avoid the obstacle. In contrast, the end effector controlled using GP-CBF successfully avoids the obstacle (see the solid blue line), which reveals the importance of safe control with learning model. Figure 5 shows the response curves of function $b(q)$ and $\psi_1(q, \dot{q})$ under the GP-CBF condition, both of which satisfy the constraints. Figures 6 and 7 are the estimates of the uncertain parts $D_a(x)$ and $\Delta(x, u)$, respectively. As shown in the plots, most of the true uncertainties (see the red lines) fall within the 95% confidence interval (see the gray fill) formed by the estimates, indicating that the estimations approximately capture the true uncertainties.

7 | CONCLUSIONS

In this article, a double-level control framework is developed for robotic systems to deal with safety-critical control with the uncertain dynamics. First, the high level planer loop plans a safe trajectory via CBF, which is used as the reference input in low level tracker loop. Afterward, the uncertainty of dynamic model is learned by GP regression method. On this basis, GP-CBF is proposed in low level tracker loop to achieve safe tracking. Then, the controller-synthesis procedure that satisfies the necessary properties of double-level safety-critical framework is investigated. Finally, the method is applied to a simulated UR3 robot to demonstrate the effectiveness.

ACKNOWLEDGMENT

This research supported by the National Natural Science Foundation of China under Grant 62173035, Grant 61803033 and Grant 61836001.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Di-Hua Zhai https://orcid.org/0000-0001-8653-8626
Yuanqing Xia https://orcid.org/0000-0002-5977-4911

REFERENCES

1. Cherubini A, Passama R, Crosnier A, Lasnier A, Fraisse P. Collaborative manufacturing with physical human–robot interaction. *Robot Comput Integr Manuf*. 2016;40:1-13.
2. Yang C, Huang D, He W, Cheng L. Neural control of robot manipulators with trajectory tracking constraints and input saturation. *IEEE Trans Neural Netw Learn Syst*. 2020;32(9):4231-4242.
3. Huang H, He W, Wang J, Zhang L, Fu Q. An all servo-driven bird-like flapping-wing aerial robot capable of autonomous flight. *IEEE/ASME Trans Mechatron*. 2022. doi:10.1109/TMECH.2022.3182418
4. Ames AD, Coogan S, Egerstedt M, Notomista G, Sreenath K, Tabuada P. Control barrier functions: theory and applications. Proceedings of the European Control Conference; 2019:3420-3431.
5. Singletary A, Klingebiel K, Bourne J, Browning A, Tokumaru P, Ames A. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems; 2021:8129-8136.
6. Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: stability and optimality. *Automatica*. 2000;36(6):789-814.
7. Prajna S, Jadbabaie A. Safety verification of hybrid systems using barrier certificates. Proceedings of the International Workshop on Hybrid Systems: Computation and Control; 2004:477-492.
8. Xu X, Waters T, Pickem D, et al. Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds. Proceedings of the IEEE Conference on Control Technology and Applications; 2017:1769-1775; IEEE.
9. Jankovic M. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*. 2018;96:359-367.
10. Lindemann L, Dimarogonas DV. Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks. *IEEE Control Syst Lett*. 2019;3(3):757-762.
11. Nguyen Q, Sreenath K. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. Proceedings of the American Control Conference; 2016:322-328.
12. Xiao W, Belta C. High order control barrier functions. *IEEE Trans Automat Contr*. 2021. doi:10.1109/TAC.2021.3105491
13. He W, Tang X, Wang T, Liu Z. Trajectory tracking control for a three-dimensional flexible wing. *IEEE Trans Control Syst Technol*. 2022. doi:10.1109/TCST.2021.3139087
14. Zhu C, Yang C, Jiang Y, Zhang H. Fixed-time fuzzy control of uncertain robots with guaranteed transient performance. *IEEE Trans Fuzzy Syst*. 2022. doi:10.1109/TFUZZ.2022.3194373
15. Yang C, Peng G, Cheng L, Na J, Li Z. Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks. *IEEE Trans Syst Man Cybern Syst*. 2019;51(5):3282-3292.
16. Nguyen Q, Sreenath K. Robust safety-critical control for dynamic robotics. *IEEE Trans Automat Contr*. 2021;67(3):1073-1088.
17. Gurriet T, Singletary A, Reher J, Ciarletta L, Feron E, Ames A. Towards a framework for realizable safety critical control through active set invariance. Proceedings of the ACM/IEEE International Conference on Cyber-Physical Systems; 2018:98-106.
18. He W, Mu X, Zhang L, Zou Y. Modeling and trajectory tracking control for flapping-wing micro aerial vehicles. *IEEE/CAA J Automat Sin*. 2021;8(1):148-156.
19. Taylor AJ, Ames AD. Adaptive safety with control barrier functions. Proceedings of the American Control Conference; 2020:1399-1405.
20. Lopez BT, Slotine JJE, How JP. Robust adaptive control barrier functions: an adaptive and data-driven approach to safety. *IEEE Control Syst Lett*. 2020;5(3):1031-1036.
21. Tanaskovic M, Fagiano L, Smith R, Morari M. Adaptive receding horizon control for constrained MIMO systems. *Automatica*. 2014;50(12):3019-3029.
22. Azimi V, Hutchinson S. Exponential control Lyapunov-Barrier function using a filtering-based concurrent learning adaptive approach. *IEEE Trans Autom Control*. 2021. doi:10.1109/TAC.2021.3120622
23. Wang N, Chen C, Yang C. A robot learning framework based on adaptive admittance control and generalizable motion modeling with neural network controller. *Neurocomputing*. 2020;390:260-267.
24. Huang H, Zhang T, Yang C, Chen CP. Motor learning and generalization using broad learning adaptive neural control. *IEEE Trans Ind Electron*. 2019;67(10):8608-8617.

25. Salehi I, Yao G, Dani AP. Active sampling based safe identification of dynamical systems using extreme learning machines and barrier certificates. Proceedings of the IEEE International Conference on Robotics and Automation; 2019:22-28.
26. Fan DD, Nguyen J, Thakker R, Alatur N, Agha-mohammadi A, Theodorou EA. Bayesian learning-based adaptive control for safety critical systems. Proceedings of the IEEE International Conference on Robotics and Automation; 2020:4093-4099; IEEE.
27. Clark A. Control barrier functions for stochastic systems. *Automatica*. 2021;130:109688.
28. Seeger M. Gaussian processes for machine learning. *Int J Neural Syst*. 2004;14(02):69-106.
29. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. Proceedings of the International Conference on Machine Learning; 2016:1050-1059.
30. Harrison J, Sharma A, Pavone M. Meta-learning priors for efficient online Bayesian regression. Proceedings of the International Workshop on the Algorithmic Foundations of Robotics; 2018:318-337.
31. Emam Y, Glotfelter P, Wilson S, Notomista G, Egerstedt M. Data-driven robust barrier functions for safe, long-term operation. *IEEE Trans Robot*. 2022;38(3):1671-1685.
32. Dhiman V, Khojasteh MJ, Franceschetti M, Atanasov N. Control barriers in Bayesian learning of system dynamics. *IEEE Trans Automat Contr*. 2021. doi:10.1109/TAC.2021.3137059
33. Blanchini F. Set invariance in control. *Automatica*. 1999;35(11):1747-1767.
34. Khalil HK. *Nonlinear systems*. Patience Hall; 2002.
35. Ames AD, Grizzle JW, Tabuada P. Control barrier function based quadratic programs with application to adaptive cruise control. Proceedings of the IEEE Conference on Decision and Control; 2014:6271-6278.
36. Cortez WS, Oetomo D, Manzie C, Choong P. Control barrier functions for mechanical systems: theory and application to robotic grasping. *IEEE Trans Control Syst Technol*. 2019;29(2):530-545.
37. Breeden J, Garg K, Panagou D. Control barrier functions in sampled-data systems. *IEEE Control Syst Lett*. 2021;6:367-372.
38. Xiao W, Belta C, Cassandras CG. Adaptive control barrier functions. *IEEE Trans Automat Contr*. 2021;67(5):2267-2281.
39. Xiao W, Belta CA, Cassandras CG. Sufficient conditions for feasibility of optimal control problems using control barrier functions. *Automatica*. 2022;135:109960.
40. Sciavicco L, Siciliano B. *Modelling and Control of Robot Manipulators*. Springer Science & Business Media; 2001.
41. Jagtap P, Pappas GJ, Zamani M. Control barrier functions for unknown nonlinear systems using Gaussian processes. Proceedings of the IEEE Conference on Decision and Control; 2020:3699-3704.
42. Taylor A, Singletary A, Yue Y, Ames A. Learning for safety-critical control with control barrier functions. *Proceedings of 2nd Conference on Learning for Dynamics and Control*; 2020:708-717.
43. Wang C, Meng Y, Li Y, Smith SL, Liu J. Learning control barrier functions with high relative degree for safety-critical control. Proceedings of the European Control Conference; 2021:1459-1464.
44. Abiko S, Yoshida K. Adaptive reaction control for space robotic applications with dynamic model uncertainty. *Adv Robot*. 2010;24(8-9):1099-1126.
45. Zheng Z, Sun L. Adaptive sliding mode trajectory tracking control of robotic airships with parametric uncertainty and wind disturbance. *J Franklin Inst*. 2018;355(1):106-122.
46. Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning. arXiv:1312.5602, 2013.

How to cite this article: Zhang S, Zhai D-H, Xiong Y, Lin J, Xia Y. Safety-critical control for robotic systems with uncertain model via control barrier function. *Int J Robust Nonlinear Control*. 2023;33(6):3661-3676. doi: 10.1002/rnc.6585