Online Control Barrier Function Construction for Safety-Critical Motion Control of Manipulators

Xuda Ding, *Student Member, IEEE*, Han Wang, Yi Ren, Yu Zheng, *Senior Member, IEEE*, Cailian Chen, *Member, IEEE*, and Jianping He, *Senior Member, IEEE*

***Abstract*—Designing safety-critical control for robotic manipulators is challenging, especially in a cluttered environment. This article proposes an online control barrier function (CBF) construction method, which extracts CBF from distance samples and enforces the safety of the motion control of robotic manipulators. Specifically, the CBF guarantees the controlled invariant property for considering the system dynamics. The proposed method samples the distance function and determines the safe set. Then, the CBF is synthesized based on the safe set by a scenario-based sum-of-square program. Unlike most existing linearization-based approaches, our method preserves the volume of the feasible space for planning without approximating the signed distance function, which helps find a solution in a cluttered environment. The control law is obtained by solving a real-time CBF-based quadratic program. Moreover, our method guarantees safety with the probabilistic result validated on a 7-DOF manipulator in real and virtual environments. The experiments show that the manipulator is able to execute tasks where the potential clearance between obstacles is in millimeters.**

***Index Terms*—Control barrier function (CBF), robotic manipulators, safe-critical control, scenario optimization, sum-of-square (SOS) program.**

I. Introduction

With the development of robotics, manipulators are broadly used in manufacturing processes and real-life [1]. Usually, there are obstacles in the working environment around the manipulators. Therefore, safety-critical motion control is fundamental for applications of robotic manipulators since manipulators need to be driven to a specified goal without collisions [2]. The whole body must have no collisions with obstacles and itself. In the past decades, path planning methods combined with tracking control have been proposed to generate a safe path for manipulators.

Encouraged by the efficient applications of search methods in path planning, rapidly exploring random trees [3] and probabilistic road maps [4] are proposed. These methods are probabilistically complete, which means they may take a long time to achieve asymptotic optimality. To deal with the optimality problem, the nonlinear optimization algorithms, such as TrajOpt [5], CHOMP [6], and OMPL [7], are used to plan joint trajectories. However, the actual trajectory might deviate from the planned one due to the complex dynamics and cluttered environments [8]. A slight trajectory deviation might cause a collision in a cluttered environment where the potential clearance between the manipulator and obstacles is less than a few millimeters, leading to unsafe.

System-dynamic-based optimization methods are proposed to deal with deviation problems. The system dynamic is considered in the constraints of the optimization. The constraints also amend that the manipulator has no collision with obstacles. In this way, motion control can satisfy the collision-free requirement. Mixed-integer linear and quadratic problems (QPs) are proposed to generate motion control by sampling and calculating the points on the manipulator's surface to the obstacles [9]. Since the number of integer variables constraints is large, these methods take seconds to minutes to solve. The QP approach is used for motion control in an online manner without integer variables [10].

In recent, with the revisiting of the controlled invariant set and control barrier function (CBF) in control theory, the CBF-based QP approach is used for safe-critical motion planning and enforces the safety [8], [11]. Different from barrier functions that bound the states (or errors of the states) of the system dynamic, CBF ensures safety by considering the intrinsic relationship between the inputs and states of the system [12]. Then, the states can be determined in a desired set. However, it is still challenging to construct constraint functions concerning collision since the manipulator's configuration space does not match the obstacles' space. Sampling-based methods are mainstream proposed to obtain the signed distance function (SDF) between two objects considering the meshes [5], [13], [14]. The SDF represents the shortest distance and gives the nearest points of two objects in the manipulators' configuration space. The differentiation of SDF is discontinuous since there is a min and max operation to obtain the function [5]. Directly using SDF in QP usually leads to a local minimum and violates the CBF-based optimization

Manuscript received 10 October 2023; revised 7 January 2024; accepted 8 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62025305, Grant 92167205, Grant 61933009, and Grant 62373247. This article was recommended by Associate Editor W. He. *(Corresponding author: Jianping He.)*

Xuda Ding, Cailian Chen, and Jianping He are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: dingxuda@sjtu.edu.cn; cailianchen@sjtu.edu.cn; jphe@sjtu.edu.cn).

Han Wang is with the Department of Engineering Science, University of Oxford, OX1 3PJ Oxford, U.K. (e-mail: han.wang@eng.ox.ac.uk).

Yi Ren is with the Advanced Manufacturing Laboratory, Huawei Technologies, Shenzhen 518129, China (e-mail: even.renyi@huawei.com).

Yu Zheng is with the Robotics X Lab, Tencent Technology, Shenzhen 518000, China (e-mail: petezheng@tencent.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2024.3387434.

Digital Object Identifier 10.1109/TSMC.2024.3387434

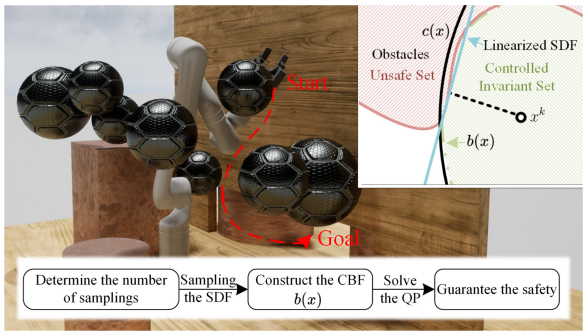

Fig. 1. Illustration of the proposed safety-critical optimal control method. The linearized SDF at x^k overlooks some feasible space. Our approach uses SDF samples to construct the control invariant set and preserves the volume of the feasible space. The subfigure on the right side is the illustration of the CBF construction. The subfigure at the bottom shows the control procedure.

requirement of continuity [8]. Constructing the constraint by linearizing the SDF is used to solve this problem [5], [8], [10]. However, the linearization of SDF leads to a small feasible space for optimization and an infeasible solution, especially when the collision meshes are complex envelopes in a cluttered environment [5]. Furthermore, the linearization of SDF or SDF determines safety, but they do not necessarily equal CBF since neither relates to the nature of the system dynamics.

Overall, it is still challenging to construct a CBF for safety-critical control of manipulators working in a cluttered environment. This article aims to design an online CBF construction control method in a cluttered environment and enforces safety when using the CBF-based QP motion control. This work considers the collision of the manipulator's whole body. Some critical issues need to be solved: 1) since the explicit SDF cannot be determined, the safe set is unknown; 2) the safe set needs to be determined based on SDF sampling and extract a candidate CBF based on a safe set; and 3) the number of SDF samples needed for estimating the safe set when constructing the CBF and the probability of safety still need to be determined. To deal with these challenges, our approach first formulates the safe set based on SDF, considering the collisions with obstacles and the manipulator itself. Then, we analyze the relationship between the safe and controlled invariant sets. The relationship provides the basic principle for CBF construction. The ellipsoidal-Lyapunov-like CBF candidate is extracted based on the safe set with online sampling, which guarantees that the control invariant set is a subset of the safe set. Furthermore, a theoretical probability guarantee of safety is given based on the number of SDF sampling used to extract the CBF. Fig. 1 illustrates the proposed safety-critical optimal control method for manipulators in a cluttered environment.

The main contributions are

1) A novel online CBF construction methodology is proposed. The novelty lies in the following.
   a) Different from constructing CBF using the linearized SDF. The safety conditions are given by analyzing the relationship between the invariant and safe sets. Then, our method constructs the CBF with the dynamic model and SDF samples with a scenario-based sum-of-square (SOS) program.
   b) A maximum optimization problem with safety conditions is used to find the CBF parameters without linearizing the SDF, which preserves the feasible space for planning.
2) Unsafety probability bound is analytically expressed, related to the number of samples and support constraints. Based on the probability, users can easily choose the SDF sampling number for CBF construction and safety-critical control.
3) Applications of the proposed method are conducted on a full-scale redundant manipulator (Kinova Gen3) in obstacle-cluttered environments. The computation speed and efficacy of the proposed method are extensively explored in real-world environments, and the method has been demonstrated to guarantee system safety and real-time performance in terms of controlling a manipulator in a cluttered environment with a clearance of millimeters.

The remainder of this article is organized as follows. Section II formulates the optimal control problem with CBF and introduces the relationship among safety, control invariant set, and the SDF. The online CBF construction method and the probability guarantee for safety are given in Section III. Simulation and implementation are shown in Section IV. Section V concludes this article.

II. Preliminaries

A. System Dynamics and Objective

Consider a dynamic model of a robotic manipulator in control-affine form as follows [8]:

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

where state $\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^n$ and input $\boldsymbol{u} \in \boldsymbol{\mathcal{U}} \subset \mathbb{R}^m$, $\boldsymbol{\mathcal{U}}$ is a compact set. f and g are locally Lipschitz continuous. Define a feedback control $\boldsymbol{u} = k(\boldsymbol{x})$, where $k : \mathbb{R}^m \to \mathbb{R}^n$ is a Lipschitz continuous control law. In this study, the state $\boldsymbol{x}$ stands for all joints' angles. The control law controls the manipulator $\mathcal{A}$ from the start position to the goal. Given the initial condition $\boldsymbol{x}(t_0) \in \mathbb{R}^n$, system (1) has a unique solution.

To perform a safety-critical motion control and design a safe input, *controlled invariant set* and CBF are used as common tools in recent studies [11], [15], [16]. Here, we first define *controlled invariant set*.

Definition 1 (Controlled Invariant Set): The set $\boldsymbol{\mathcal{B}} \subset \boldsymbol{\mathcal{X}}$ is a controlled invariant set for system (1) if for every initial condition $\boldsymbol{x}(t_0) \in \boldsymbol{\mathcal{B}}$, there exists an input $\boldsymbol{u}(t)$ such that $\boldsymbol{x}(t) \in \boldsymbol{\mathcal{B}}$ for time $t \in [t_0, t_{\max})$. When $t_{\max} = \infty$, system (1) is *forward complete*.

The *controlled invariant set* guarantees the states of the manipulator flow in a region with respect to the system dynamics [15], [17]. Roughly speaking, the states must be in the set $\boldsymbol{\mathcal{B}}$ and never leave. This motivates the formulation of CBFs [11], [16]. Here, we introduce an extended class $\mathcal{K}_\infty$ function α, which is continuous, strictly monotonically increasing, and satisfies $\alpha(0) = 0$, $\lim_{i\to\infty} \alpha(i) = \infty$ and $\lim_{i\to-\infty} \alpha(i) = -\infty$.

Definition 2 (CBF): Let $\boldsymbol{\mathcal{B}} \subset \boldsymbol{\mathcal{X}}$ be the zero super-level set of a continuously differentiable function $b : \mathcal{X} \to \mathbb{R}$, then b is a CBF if there exists an extended class $\mathcal{K}_\infty$ function α such that for the control system (1)

$$\forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}, \sup_{\boldsymbol{u}\in\boldsymbol{\mathcal{U}}} \underbrace{\left[\frac{\partial b}{\partial \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\partial b}{\partial \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{u}\right]}_{\dot{b}(\boldsymbol{x},\boldsymbol{u})} \geq -\alpha(b(\boldsymbol{x})) \tag{2}$$

where $\dot{b}(\boldsymbol{x}, \boldsymbol{u})$ is the derivative of $b(\boldsymbol{x})$ along system (1).

Now we define a set $\mathcal{K}_c(\boldsymbol{x})$, which consists of all control values that make the states of the manipulator flow in $\boldsymbol{\mathcal{B}}$

$$\mathcal{K}_c(\boldsymbol{x}) = \left\{\boldsymbol{u} \in \boldsymbol{\mathcal{U}} : \dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x})) \geq 0\right\}. \tag{3}$$

When the control input $k(\boldsymbol{x}) \in \mathcal{K}_c(\boldsymbol{x})$, the state $\boldsymbol{x} \in \boldsymbol{\mathcal{B}}$. As it is pointed out in [8], since the complex dynamics and cluttered environment, it may be the case that $k(\boldsymbol{x}) \notin \mathcal{K}_c(\boldsymbol{x})$ for some $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$, and causes $\boldsymbol{x} \notin \boldsymbol{\mathcal{B}}$. Moreover, when the outside of $\boldsymbol{\mathcal{B}}$ is the obstacle, the collision will happen (formal definition of *safe set* and its relationship with $\boldsymbol{\mathcal{B}}$ will be given in the following section). For example, obstacles may shift locations, and the manipulators may initially have slightly different joint configurations. These situations may cause the previously computed control law not to drive the manipulator as desired and cause a collision. Given a control law k, to guarantee $\boldsymbol{u} \in \mathcal{K}_c(\boldsymbol{x})$ and further guarantee $\boldsymbol{x} \in \boldsymbol{\mathcal{B}}$, a modification of the control law k in a minimal way is given

$$\begin{aligned} \boldsymbol{u}^*(\boldsymbol{x}) = \arg\min_{u} \ & \frac{1}{2}\|\boldsymbol{u} - k(\boldsymbol{x})\|^2 \\ \text{subject to } & \dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x})) \geq 0 \quad \text{(SCB-QP)} \end{aligned}$$

where $\boldsymbol{u}^*(\boldsymbol{x})$ is the filtered input. Notice that the conditions given in (3) are affine in $\boldsymbol{u}$, the CBF b is incorporated into the QP to synthesize point-wise optimal control law [8], [11]. The constraint in (SCB-QP) enables the filtered input $\boldsymbol{u}^*(\boldsymbol{x})$ navigate the system state in the set $\boldsymbol{\mathcal{B}}$. Equation (SCB-QP) can be solved in real-time for nonlinear systems. The *controlled invariant set* $\boldsymbol{\mathcal{B}}$ defines the forward invariance of the system (1) but does not guarantee the safety directly. The safe set needs to be defined to determine the safety considering obstacles that cause the collision.

B. Safety and SDF

The collision-free situations require the clearance distance between the manipulator and the obstacle is always non-negative. Here, we define SDF $h : \mathbb{R}^n \to \mathbb{R}$ to quantify the minimum distance between the manipulator and the obstacles (including the different links of the manipulator, which will be discussed in Section III-A). The manipulator does not collide when $h(\boldsymbol{x}) \geq 0$. Otherwise, the collision (penetration) happens and leads to unsafe. This allows us to define *safe set*.

Definition 3 (Safe Set): The set $\boldsymbol{\mathcal{S}} \subset \boldsymbol{\mathcal{X}}$ is a safe set for system (1) if there exists a SDF $h : \mathcal{X} \to \mathbb{R}$

$$\boldsymbol{\mathcal{S}} = \{\boldsymbol{x} \in \boldsymbol{\mathcal{X}} | h(\boldsymbol{x}) \geq 0\}. \tag{4}$$

When $\boldsymbol{\mathcal{B}} \subseteq \boldsymbol{\mathcal{S}}$, the *controlled invariant set* $\boldsymbol{\mathcal{B}}$ is safe, and the filtered input $\boldsymbol{u}^*$ guarantees the intrinsic safety. On the contrary, $\boldsymbol{u}^*$ cannot guarantee safety when only a subset of $\boldsymbol{\mathcal{B}}$ is in the safe set. CBF sets the system input in $\mathcal{K}_c(\boldsymbol{x})$ and then guarantees the states in $\boldsymbol{\mathcal{B}}$, while barrier functions used in [12] and [18] set a potential field in states to avoid collision. CBF guarantees the states in the controlled invariant set and enforces safety. The following lemma promises the existence of such controller $\boldsymbol{u}^*$.

Lemma 1: There exists $\boldsymbol{u}^*$ such that system (1) is able to maintain safety under $\boldsymbol{\mathcal{S}}$, if and only if there exists a controlled invariant set $\boldsymbol{\mathcal{B}} \subseteq \boldsymbol{\mathcal{S}}$.

Remark 1: It should be noted that the SDF h used for establishing a safe set is not necessarily the same as the CBF b, for 1) The safe set constructed based on the obstacles may not guarantee the invariance property with respect to dynamics (1); 2) The SDF h may not be continuously differentiable [5].

Then, we consider how to obtain the signed distance between two objects. The meshes of the obstacles and the manipulator are considered for minimizing the workspace reduction. According to [5], [19], and [20], define $\text{dist}(\mathcal{A}, \mathcal{O}_i)$ is the distance between $\mathcal{A}$ and the obstacle $\mathcal{O}_i$, where $i = 1, \ldots, M$ and $M \in \mathbb{Z}^+$, which can be obtained as

$$\text{dist}(\mathcal{A}, \mathcal{O}_i) = \min_{\boldsymbol{d}}\{\|\boldsymbol{d}\| : (\mathcal{A} + \boldsymbol{d}) \cap \mathcal{O}_i \neq \emptyset\} \tag{5}$$

where $\boldsymbol{d}$ is the posture of the manipulator, which is related to the state $\boldsymbol{x}$. Equation (5) aims to find the minimum distance of the posture transformation of the manipulator to reach the obstacle. Similarly, to consider the negative distance, penetration distance is defined as

$$\text{pen}(\mathcal{A}, \mathcal{O}_i) = \min_{\boldsymbol{d}}\{\|\boldsymbol{d}\| : (\mathcal{A} + \boldsymbol{d}) \cap \mathcal{O}_i = \emptyset\}. \tag{6}$$

Then, with $\text{dist}(\mathcal{A}, \mathcal{O}_i)$ and $\text{pen}(\mathcal{A}, \mathcal{O}_i)$, the signed distance is obtained following a simple way:

$$\text{sd}(\mathcal{A}, \mathcal{O}_i) = \text{dist}(\mathcal{A}, \mathcal{O}_i) - \text{pen}(\mathcal{A}, \mathcal{O}_i). \tag{7}$$

The safe set is determined with $h(\boldsymbol{x}) = \min_{i=1,\ldots,M} \text{sd}(\mathcal{A}, \mathcal{O}_i)$ when only considering the collision between the manipulator and obstacles. We direct readers to [5], [13], and [14] for detailed information about the SDF. When global information cannot be obtained, the obstacles need to be sampled and constructed. *Voxblox* [21] and *FIESTA* [22] can be used to incrementally construct the euclidian signed distance fields (ESDFs) of the obstacles and the signed distance $\text{sd}(\mathcal{A}, \mathcal{O}_i)$ is obtained.

The SDF is nonsmooth and not continuously differentiable in general, and its explicit form is hard to obtain. Linearization of SDF is used for convex formulation in [5], [8], and [23]. However, such linearization could overlook some space and lead to a small feasible space and infeasible solutions for motion control, especially in a cluttered environment [5]. The linearization method produces a set defined by the half plane over the state space. Such a set has no control invariance guarantee with respect to a specific vector field. Besides, using linearized SDF as CBF is illogical in general. CBF indicates the invariant property of the system, and SDF gives information about the environment. This motivates us to construct CBF based on SDF rather than using the linearized SDF to construct CBF in (SCB-QP) for safety-critical motion control in a cluttered environment.

The following difficulties are tackled in this article to construct CBF and synthesize the point-wise safety-critical control.

1) To formulate a general SDF and determine the *safe set*, besides the collision between the manipulator and obstacles, the collision between different manipulator links needs to be considered.
2) Since the explicit SDF cannot be obtained, where to sample the SDF and how to design a CBF that makes $\boldsymbol{\mathcal{B}} \subseteq \boldsymbol{\mathcal{S}}$ with reserving feasible space for optimal control need to be studied.
3) The probability of $\boldsymbol{\mathcal{B}} \subseteq \boldsymbol{\mathcal{S}}$ needs to be determined since the number of SDF samples is finite and precise SDF is unknown.

III. Online CBF Construction Method

This section shows the main results of our online CBF construction method based on SDF sampling. The samples are used because the explicit form of SDF cannot be obtained. This section is organized in three parts. In Section III-A, we reformulate a general SDF considering two kinds of collisions, i.e., outer collision (obstacles and the manipulator) and inner collision (different links of the manipulator). In Section III-B, the CBF construction method considering the *safe set* with the SOS program is proposed, and where to sample SDF is determined. Based on the samples, the online CBF construction method is given with the probabilistic guarantee for safety in Sections III-C and III-D.

A. SDF Determination

1) Outer Collision: The direct interpretation of collision-free between the manipulator $\mathcal{A}$ and environmental obstacles $\{\mathcal{O}_i\}$ is $\mathrm{sd}(\mathcal{A}, \mathcal{O}_i) \geq 0, \forall i \in \mathcal{I}$. However, since it has no explicit expression, this condition is intractable to use as a constraint in motion control optimization problems. Formally, the outer-SDF is defined as

$$\mathrm{sd}_{\mathrm{out}}(\boldsymbol{x}) = \max_{\|\hat{\boldsymbol{n}}_{\mathrm{out}}\|_2=1} \min_{\substack{p_{\mathcal{A}} \in \mathcal{A} \\ p_{\mathcal{O}_i} \in \mathcal{O}_i}} \hat{\boldsymbol{n}}_{\mathrm{out}} \cdot \left(F^w_{\mathcal{A}}(\boldsymbol{x}) p_{\mathcal{A}} - F^w_{\mathcal{O}_i}(\boldsymbol{x}) p_{\mathcal{O}_i} \right) \tag{8}$$

where $\hat{\boldsymbol{n}}_{\mathrm{out}}$ is the direction of the minimal translation $\boldsymbol{d}$ in (5) and (6). $p_{\mathcal{A}}$, $p_{\mathcal{O}_i}$ are two points on the manipulator and the obstacle in their local coordinates, and they are needed to be transferred into world coordinates. These two points in the world coordinates are $F^w_{\mathcal{A}}(\boldsymbol{x}) p_{\mathcal{A}} \in \mathbb{R}^3$ and $F^w_{\mathcal{O}_i} p_{\mathcal{O}_i} \in \mathbb{R}^3$. The function F^w is forward kinematics, which gives the pose of the manipulator and the obstacles in the world frame. $\mathrm{sd}_{\mathrm{out}}$ can be obtained by sampling points on the controlled object and obstacles, and hereafter using the GJK [13] or EPA [14] algorithm. With the amount of data in the magnitude of hectobit, the function can be constructed implicitly within milliseconds.

2) Inner Collision: In addition to the outer collision, another possible collision scenario happens for different links of the manipulator. For this case, the inner-SDF is defined by

$$\mathrm{sd}_{\mathrm{in}}(\boldsymbol{x}) = \max_{\|\hat{\boldsymbol{n}}_{\mathrm{in}}\|_2=1} \min_{\substack{p_{\mathcal{A}} \in \mathcal{A} \\ p'_{\mathcal{A}} \in \mathcal{A}}} \hat{\boldsymbol{n}}_{\mathrm{in}} \cdot \left(F^w_{\mathcal{A}}(\boldsymbol{x}) p_{\mathcal{A}} - F^w_{\mathcal{A}}(\boldsymbol{x}) p'_{\mathcal{A}} \right) \tag{9}$$

where $p_{\mathcal{A}}$ and $p'_{\mathcal{A}}$ are different points on the different links of the manipulator. We define the overall-SDF as

$$h(\boldsymbol{x}) = \min\{\mathrm{sd}_{\mathrm{out}}(\boldsymbol{x}), \mathrm{sd}_{\mathrm{in}}(\boldsymbol{x})\} \tag{10}$$

where the composition relationship in $\mathrm{sd}_{\mathrm{ov}}$ is captured by a $\wedge$ quantifier. If for a $\boldsymbol{x}$, the manipulator is both inner-collision free and outer-collision free, then $\mathrm{sd}_{\mathrm{in}}(\boldsymbol{x}) \geq 0 \wedge \mathrm{sd}_{\mathrm{out}}(\boldsymbol{x}) \geq 0$, which is equivalent to $h(\boldsymbol{x}) \geq 0$. When $h(\boldsymbol{x}) = 0$, the manipulator is at the boundary of collision; when $h(\boldsymbol{x}) > 0$, the manipulator is away from collision; when $h(\boldsymbol{x}) < 0$, the collision happens. Then, the safe set of the manipulator is constructed with h.

B. CBF Construction Considering the Safe Set

In most of the existing literature, the SDF is directly used or linearized and then used as a CBF in safety-critical controller design problems [8], [23], [24]. However, the safe set $\boldsymbol{\mathcal{S}}$ defined by the zero-super level set of distance function is unnecessary to be a CBF candidate, as it has no controlled invariance property. The whole construction procedure in Section III-A does not utilize the manipulator's dynamic model (1). This section considers the dynamics and shows how to synthesize a CBF b from the safe set $\boldsymbol{\mathcal{S}}$ based on SDF h.

Lemma 2: For manipulator $\mathcal{A}$ with dynamics (1), and SDF h, b is a CBF candidate and guarantees the safety of $\mathcal{A}$ if

$$b(\boldsymbol{x}) \leq h(\boldsymbol{x}) \tag{11a}$$

$$\forall \boldsymbol{x} \in \partial\boldsymbol{\mathcal{B}}, \exists \boldsymbol{u} \in \boldsymbol{\mathcal{U}}, \quad \dot{b}(\boldsymbol{x}, \boldsymbol{u}) \geq 0. \tag{11b}$$

Proof: It is a sufficient condition for safety. The inequality in (11a) stands for the zero super-level set of the CBF b is a subset or equals to that of the SDF h, i.e., when $b(\boldsymbol{x}) = 0$, $h(\boldsymbol{x}) \geq 0$. Under this property, if $b(\boldsymbol{x}) \geq 0$ for any $\boldsymbol{x}$ on the trajectory, the invariance can be guaranteed with (11b), and then $h(\boldsymbol{x}) \geq 0$ which guarantees safety. Condition (11b) is a standard controlled invariance condition [11]. ■

Without an additional objective, the construction of b leads to the following feasibility optimization problem:

$$\begin{aligned} &\text{find } b(\boldsymbol{x}) \\ &\text{subject to (11).} \end{aligned} \tag{12}$$

Although the conditions are elegant in terms of algebraic structure, it is still very hard to construct a CBF b by solving (12). The challenges here are 1) Conditions (11a) and (11b) should hold for $\forall \boldsymbol{x} \in \boldsymbol{\mathcal{X}}$, which renders (12) to be an infinitely constrained optimization problem and 2) h has only an implicit form since it is a composition of solutions to two optimization problems in (8) and (9). This section will tackle all these challenges, but we want to point out that the computational complexity of (12) is still high. The reason is that the explicit h cannot be obtained. Estimation of h based on a large number of samplings is time-consuming. The real-time computing problem will be fixed with a relaxation method in

Section III-C. Now, we will determine the reachable set $\mathcal{C}$ where the samples are obtained to construct CBF b.

1) Reachable Set Determination: We suppose that the current state of the manipulator is $\boldsymbol{x}^k$. Given that the safe set $\mathcal{S}$ is constructed from the sampled data online, the CBF b should also be synthesized online. The definition domain $\mathcal{C}$ is therefore varying with the state $\boldsymbol{x}$. Starting from this point, the maximum movement $||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k}$ of the manipulator is

$$||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k} = \max_{\boldsymbol{u}\in\mathcal{U}} \left\| \left\{ f\left(\boldsymbol{x}^k\right) + g\left(\boldsymbol{x}^k\right)\boldsymbol{u} \right\} dt \right\|^2 \tag{13}$$

where dt is the control (sampling) time interval. In this way, the state is ensured within the safe set during the solution's control (sampling) interval. The set $\mathcal{C}$ at state $\boldsymbol{x}^k$ is then defined by a ball $\mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$ centered on $\boldsymbol{x}^k$, with radius $||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k}$. The reason why we use a high-dimensional ball but not the exact reachable region, i.e., $\bigcup_{\boldsymbol{u}\in\mathcal{U}}\{f(\boldsymbol{x}^k) + g(\boldsymbol{x}^k)\boldsymbol{u}\}dt$, is that $\mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$ has a good convexity and is computationally cheaper. Clearly, $\bigcup_{\boldsymbol{u}\in\mathcal{U}}\{f(\boldsymbol{x}^k) + g(\boldsymbol{x}^k)\boldsymbol{u}\}dt \subseteq \mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$.

2) Construction of the CBF: Then, the CBF is constructed in the following form:

$$b(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{H}_b \boldsymbol{x} + d_b \tag{14}$$

where $\boldsymbol{H}_b \prec 0$, and $d_b > 0$. This kind of CBF originates from the quadratic Lyapunov function $v(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{H}_b \boldsymbol{x}$ [25], where $v{:}\mathcal{X} \to \mathbb{R}$. Then, we have $b(\boldsymbol{x}) = d_b - v(\boldsymbol{x})$. Parameterizing the CBF b to be ellipsoidal is motivated by the ellipsoidal Lyapunov function [16]. For a stabilizable linear system, it is reasonable to consider an ellipsoidal controlled invariant set, which is the complement set of the super-level set of an ellipsoidal Lyapunov function. In our application, when the control frequency is high, i.e., dt is small, the Lebesgue measure of the ball $\mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$ would be relatively small. In such a small region, the original nonlinear system (1) can be linearized with a slight bias around $\boldsymbol{x}^k$. Then, the CBF b can be used to synthesize the point-wise optimal control (SCB-QP).

With this quadratic parameterization, we can use SOS relaxation and S-procedure [26] to guarantee that $\mathcal{B} \subseteq \mathcal{C}$, as the controlled invariant set is inside the current reachable set

$$-b(\boldsymbol{x}) + \sigma_1 c(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \tag{SOS-CBF}$$

where $\sigma_1 \in \Sigma[\boldsymbol{x}]$ is an SOS multiplier, and $c(\boldsymbol{x}) = -||\boldsymbol{x} - \boldsymbol{x}^k||^2 + ||\delta \boldsymbol{x}||^2_{\boldsymbol{x}=\boldsymbol{x}^k}$. We recall that the set $\mathcal{C}$ is restricted to be a ball centered at $\boldsymbol{x}^k$, i.e., $\mathcal{BA}(x^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$ in the first step. It is evident that c is a polynomial function. Thus, $\mathcal{C}$ is a semi-algebraic set. Together with the quadratic function b and the SOS polynomial multiplier σ_1, the SOS constraint (SOS-CBF) can be converted to a semi-definite constraint. We note here that the multiplier σ_1 will appear as an additional variable in the following synthesis optimization problem (SCSOS-CBF). Table I illustrates the different sets to provide a clear understanding.

TABLE I
ILLUSTRATION OF SET $\mathcal{S}$, $\mathcal{C}$, AND $\mathcal{B}$

Set	Denotation	Remark
$\mathcal{S}$	safe set	Defined by (4), is not explicitly applicable to the manipulator.
$\mathcal{C}$	reachable set	Defined by $\mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$, is the maximum reachable set of system (1) at $\boldsymbol{x}^k$.
$\mathcal{B}$	controlled invariant set	Synthesized by (SCSOS-CBF), and satisfies: $\mathcal{B} \subseteq \mathcal{C}$, $\mathcal{B} \subseteq \mathcal{X}$.

C. Online CBF Construction Based on SDF Samplings

The following sections show how to construct $b(\boldsymbol{x})$ for the manipulator satisfying the residual constraints in (11). In addition to the SOS synthesis approach in the last part, we use scenario optimization to alleviate some constraints. The reason why SOS is not fully applicable for the remaining constraint $\forall \boldsymbol{x} \in \mathcal{C}, b(\boldsymbol{x}) \leq h(\boldsymbol{x})$ is that h is not a polynomial function in general. More precisely, there is even no explicit expression of it by hand. Although there are lifting methods [27] and Schur relaxation methods [28] to overcome these issues, they either do not scale well with dimension or require iterative solutions. In the collision-avoidance problems for manipulators, real-time computation is rather important. This makes us turn to instead using probabilistic CBF conditions with sampled scenarios.

The scenario optimization relies on sampled scenarios to relax the original problem. Sampling all SDF with $\boldsymbol{x} \in \mathcal{C}$ and solving (SOS-CBF) is impractical since the number of samplings is infinite. Instead, we sample finite $\bar{N}$ realizations of $\boldsymbol{x}^{(r)}$ around $\boldsymbol{x}^k$. The samples are with a probability measure π, which satisfies

$$\int_{\mathcal{C}} \pi(\boldsymbol{x}) d\boldsymbol{x} = 1. \tag{15}$$

Let $\bar{\boldsymbol{X}} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\bar{N})}\}$ be the set of sampled scenarios. These scenarios are independently and identically sampled according to π. Then, the SDF is sampled form the scenarios and $h(\boldsymbol{x}^{(i)})$ is obtained, where $i = 1, 2, \ldots, \bar{N}$. We could construct the following scenario feasibility program:

$$\begin{aligned}
&\underset{\boldsymbol{u}\in\mathcal{U}}{\text{find}} \quad b(\boldsymbol{x}) && \text{(16a)}\\
&\text{subject to } b\left(\boldsymbol{x}^{(i)}\right) \leq h\left(\boldsymbol{x}^{(i)}\right) && \text{(16b)}\\
&\qquad \dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x}))\big|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}} \geq 0 && \text{(16c)}\\
&\qquad \forall\, \boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}}. && \text{(16d)}
\end{aligned}$$

The lower-envelope condition $\forall \boldsymbol{x} \in \mathcal{C}, b(\boldsymbol{x}) \leq h(\boldsymbol{x})$ is enforced only on the finite set of scenarios $\bar{\boldsymbol{X}}$. The controlled invariance condition (11b) is substituted by a relaxed formulation $\forall \boldsymbol{x} \in \mathcal{B}, \exists \boldsymbol{u} \in \mathcal{U}, \dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x})) \geq 0$. The relaxed formulation leads to a convex problem, which helps find a numerical solution. Moreover, the function α can be tuned to help obtain the solution for (16) as a relaxation term.

Notice that the equivalence between $b(x(i)) \leq h(x(i))$ and $b(x) \leq h(x)$ cannot be guaranteed since the number of sampling is finite. This leads us to formulate a CBF b from the SDF h samplings form $\bar{\boldsymbol{X}}$ with a probability of safety $1-\epsilon$ where $\epsilon \in [0, 1)$.

Theorem 1: For manipulator $\mathcal{A}$ with dynamics (1) at the current state $\boldsymbol{x}^k$. Let $\bar{\boldsymbol{X}} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\bar{N})}\}$ be a set of samplings in reachable set $\mathcal{C}$, $h(\boldsymbol{x}^{(i)})$ is the samplings of SDF

Algorithm 1: Online CBF Construction

Input: the number of samples $\bar{N}$, current state $\boldsymbol{x}^k$, the maximum ball set $\mathcal{BA}(\boldsymbol{x}^k, ||\delta \boldsymbol{x}||_{\boldsymbol{x}=\boldsymbol{x}^k})$.

Output: CBF parameter $\boldsymbol{H}_b$ and d_b.

1 Initialize the SOS program according to (SCSOS-CBF) and $\bar{N}$.
2 Randomly generate $\bar{N}$ samples according to $\pi(\boldsymbol{x})$
3 **for** $i \leq \bar{N}$ **do**
4 | Compute $h(\boldsymbol{x}^{(i)})$ for all samples in $\bar{X}$.
5 **end**
6 Solve the optimal problem (SCSOS-CBF).
7 Return $\boldsymbol{H}_b$ and d_b.

where $i = 1, 2, \ldots, \bar{N}$, then b is a CBF and guarantees $\boldsymbol{\mathcal{B}} \subseteq \boldsymbol{\mathcal{S}}$ with probability $1 - \epsilon$ if

$$\begin{aligned} &\forall\, \boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}}, \quad b\Big(\boldsymbol{x}^{(i)}\Big) \leq h\Big(\boldsymbol{x}^{(i)}\Big) \\ \forall \boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}}, \exists \boldsymbol{u} \in \boldsymbol{\mathcal{U}}, \quad &\dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x}))\big|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}} \geq 0 \\ &- b(\boldsymbol{x}) + \sigma_1 c(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \\ &\sigma_1 \in \Sigma[\boldsymbol{x}]. \end{aligned} \tag{17}$$

Proof: Condition $\forall\, \boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}}$ restricts the sampling points in the reachable set $\boldsymbol{\mathcal{C}}$. Condition $b(\boldsymbol{x}^{(i)}) \leq h(\boldsymbol{x}^{(i)})$ indicates that the control invariant set is a subset of the safe set in the sense of scenarios $\boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}}$. The condition $.\dot{b}(\boldsymbol{x}, \boldsymbol{u}) + \alpha(b(\boldsymbol{x}))|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}} \geq 0$, leads to a convex problem when seeking barrier functions with numerical means. Condition $-b(\boldsymbol{x}) + \sigma_1 c(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]$ indicates that for any $\boldsymbol{x}$, $-b(\boldsymbol{x}) + \sigma_1 c(\boldsymbol{x}) \geq 0$ and further $\forall\, \boldsymbol{x} \in \boldsymbol{\mathcal{B}}$, $c(\boldsymbol{x}) \geq 0$. The analysis and proof of the safety probability are given in Section III-D. ■

The final synthesis program for the online CBF construction with SDF samples for manipulators is given as follows:

$$\begin{aligned} \underset{\boldsymbol{H}_b \prec 0, d_b > 0}{\arg\max} \quad & d_b \\ \text{subject to} \quad & (17) \\ & ||\boldsymbol{H}_b||_2 \leq 1. \end{aligned} \tag{SCSOS-CBF}$$

The constraint $||\boldsymbol{H}_b||_2 \leq 1$ is the regularization term. The objective is to maximize the value of d_b to preserve the volume of the control invariant set $\boldsymbol{\mathcal{B}}$, under the regularization of $\boldsymbol{H}_b$. Here, we note that our method does not require parameterizing the controller, unlike the results in [28]. Thus, manipulators can be controlled in a cluttered environment efficiently. Moreover, the term $\alpha(b(\boldsymbol{x}))$ can be substituted by $\alpha(b(\boldsymbol{x})) = \kappa b(\boldsymbol{x})$, where $\kappa > 0$, and using the regularization constraint $||\boldsymbol{H}_b||_F \leq 1$ for convenience of computing.

Algorithm 1 shows the online CBF construction method considering the manipulator dynamics and obstacles. Once the SOS program is built, we can solve (SCSOS-CBF) by a semi-definite programming solver with a centering-predictor-corrector Algorithm [29]. When the next state $\boldsymbol{x}^{k+1}$ comes in, we only go through lines 2–7 in Algorithm 1.

D. Probabilistic Guarantee for Safety

Since the number of sampling is finite, the online method cannot guarantee the equivalence between $b(\boldsymbol{x}^{(i)}) \leq h(\boldsymbol{x}^{(i)})$ and $b(\boldsymbol{x}) \leq h(\boldsymbol{x})$. Assume that z is the solution of (SCSOS-CBF). It would be possible that z is not in the safe solution set $\mathcal{S}_x$, and the manipulator is controlled to collision due to the uncertainty caused by sampling. To quantify the probability of obtaining unsafe solution, the *violation probability* is given as [30].

Definition 4 (Violation Probability): The *violation probability* of a given solution z is defined as $V(z) = \mathbb{P}\{\boldsymbol{x} \in \boldsymbol{\mathcal{C}} : z \notin \mathcal{S}_x\}$.

Recent results point out that the violation probability $V(z)$ is closely related to both the number of scenarios and the *complexity*, i.e., the number of *support constraints* [31].

Definition 5: A constraint in $\mathcal{S}_x$ of the synthesis program (SCSOS-CBF) is called a *support constraint* if its removal (while all the other constraints are maintained) changes the optimal solution. The *complexity* $c^*_{\bar{N}}$ of the synthesis scenario program (SCSOS-CBF) is the number of the support constraints.

Note that the decision variables $\boldsymbol{H}_b, d_b$ in (SCSOS-CBF) are stacked by $z \in \mathbb{R}^e$ for the ease of following theoretic analysis, where $e = n^2 + 1$ is the dimension of the solution. Then, the probabilistic result of violation probability based on the $\bar{N}$ scenarios is given in the following results.

Theorem 2: Consider the online CBF construction program (SCSOS-CBF) with complexity $c^*_{\bar{N}}$, given confidence parameter $\beta \in (0, 1)$, for the data sampled from $\boldsymbol{\mathcal{C}}$ satisfies (15), then the violation probability $V(z)$ is quantified as

$$\mathbb{P}^{\bar{N}}\left\{\underline{\epsilon}\left(c^*_{\bar{N}}\right) \leq V(z) \leq \bar{\epsilon}\left(c^*_{\bar{N}}\right)\right\} \geq 1 - \beta \tag{18}$$

where $\underline{\epsilon}(c^*_{\bar{N}})$ and $\bar{\epsilon}(c^*_{\bar{N}})$ are obtained based on the following equation in the ξ variable:

$$\binom{\bar{N}}{k}\xi^{\bar{N}-k} - \frac{\beta}{2\bar{N}}\sum_{i=k}^{\bar{N}-1}\binom{i}{k}\xi^{i-k} - \frac{\beta}{6\bar{N}}\sum_{i=\bar{N}+1}^{4\bar{N}}\binom{i}{k}\xi^{i-k} = 0 \tag{19}$$

where $\bar{N} > e = n^2 + 1$, and the two solutions in $[0, +\infty)$ which are denoted by $\underline{\xi}(k)$ and $\bar{\xi}(k)$, respectively. $\underline{\xi}(k) \leq \bar{\xi}(k)$. $\underline{\epsilon}(c^*_{\bar{N}}) = \max\{0, 1 - \bar{\xi}(c^*_{\bar{N}})\}$ and $\bar{\epsilon}(c^*_{\bar{N}}) = 1 - \underline{\xi}(c^*_{\bar{N}})$.

Proof: First, we determine the dimension of the solution of (SCSOS-CBF). The decision variables are $\boldsymbol{H}_b, d_b$, which could cause $b(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{H}_b \boldsymbol{x} + d_b > h(\boldsymbol{x})$ due to the uncertainty in sampling and leads to unsafe. Thus, the dimension is n^2+1. Then, as for the detailed proof of the general formulation (18) and (19), we direct the readers to [31, Th. 1]. ■

This result shows the relationship between the number of support constraints $c^*_{\bar{N}}$, the violation probability on the optimal solution $V(z)$, and parameter β. The scenario constraints are more prone to be violated if the complexity is high. One intuitive interpretation of this result is that the higher complexity is, the more boundary of constraints the sorted solution stands on. Then, the uncertain constraints have a higher risk. The following corollary is a direct result from Theorem 2.

Corollary 1: Given β, it always holds that $\underline{\epsilon}(c^*_{\bar{N}}) \leq \underline{\epsilon}(e) \leq \bar{\epsilon}(c^*_{\bar{N}}) \leq \bar{\epsilon}(e)$. Besides, we certainly have $\mathbb{P}^{\bar{N}}\{V(z^*) \leq \bar{\epsilon}(e)\} \geq 1 - \beta$.

Algorithm 2: Determine the Number of Samples

Input: the dimension of the decision variables e, confidence parameter β, violation probability level ϵ, the predefined maximum number $\bar{N}_{\max}$, probability threshold ϵ_p.

Output: the minimum number of the sample $\bar{N}$.

```
1  N̄_min = e.
2  while N̄_min + 1 ≤ N̄_max do
3      compute the inverse incomplete beta function → γ_L
       and → γ_U, based on (β, e, N̄_min − e + 1) and
       (β, e, N̄_max − e + 1), respectively.
4      t_L1 = 1 − γ_L, t_L2 = 1, t_U1 = 1 − γ_U, t_U2 = 1.
5      obtain P_L1 P_L2 P_U1 and P_U2 according to (19) based
       on t_L1, t_L2, t_U1 and t_U2.
6      if P_L1 × P_L2 ≥ 0 then
7          ε_1 = 0.
8      else
9          while t_L2 − t_L1 > 0 do
10             t_L = ⌈(t_L2 + t_L1)/2⌉.
11             obtain P_tL according to (19) based on t_L.
12             if P_tL > 0 then
13                 t_L1 = t_L.
14             else
15                 t_L2 = t_L.
16             end
17             ε_1 = 1 − t_L2.
18         end
19     end
20     compute ε_2 based on t_U1, t_U2, P_U1 and P_U2.
21     if |ε_1 − ε| > ε_p or |ε_1 − ε_2| > ε_p then
22         N̄_min = N̄_min + ⌈(N̄_min − N̄_max)/2⌉
23     else
24         N̄ = N̄_min.
25         return N̄
26     end
27 end
```

One direct application of this result is that we can measure how many samples are required for a given confidence parameter β and violation probability level ϵ for manipulators. The following lemma concludes the amount of data.

Lemma 3: Given violation probability level $\epsilon \in [0, 1)$, confidence parameter $\beta \in [0, 1)$, then the amount $\bar{N}(\epsilon, \beta)$ of samples required to render $\mathbb{P}^{\bar{N}}\{V(z^*) \leq \epsilon\} \geq 1 - \beta$, where $\epsilon = \bar{\epsilon}(e)$ fulfills

$$\bar{N}(\epsilon, \beta) \geq \begin{cases} \arg\min \ \bar{N} \\ \bar{N} \in \mathbb{N} \\ \text{s.t. (19).} \end{cases} \tag{20}$$

Although Lemma 3 gives guidance on how many sampled data are necessary for the acceptable violation probability level and confidence, the result is hard to obtain since the optimization problem in (20) is a nonconvex mixed integer program. We provide a heuristic algorithm which can compute $\bar{N}$ given e, ϵ and β. There are two levels of the dichotomic search program in Algorithm 2. The first one computes the minimum number of samples. The second one computes the violation probability. Line 21 gives the terminal conditions: 1) the risk based on $\bar{N}$ samples is near the risk goal and 2) the risk cannot decrease too much with $\bar{N}$ increasing.

Furthermore, one of the prominent applications of our proposed method is its utilization as an online filter for predefined trajectory path planning. The trajectory itself can be generated using various path planning algorithms, such as A* [32], RRT [33], and PRM [34]. These algorithms facilitate the generation of offline way-points that define the desired path. Once these way-points are determined, our method can be employed in real-time to ensure the system's safety. In Section IV, we will introduce a similar control synthesis technique specifically designed for filtering a predefined trajectory. The only distinction is that our trajectory consists of seven dimensions, setting it apart from conventional path-planning approaches.

Regarding the adaptability of different dimensions, the proposed method demonstrates applicability to systems with varying state dimensions from a mathematical standpoint. However, in practical implementation, increased state dimensions result in longer computational time required to solve (SCSOS-CBF). Additionally, higher-state dimensions necessitate increased sampling of SDF to maintain violation probabilities at desired levels.

IV. Simulation and Experimental Implementation

This section gives the control synthesis with the proposed CBF construction method to formulate a safety-critical control for the robotic manipulator. The Simulation and Experimental Implementation aim to show

1) The efficacy of the CBF construction method in a simple linear system from a numerical example.
2) The computation time of the online CBF construction method under different violation probabilities and tolerance of the solver.
3) The efficacy of the proposed CBF construction method for robotic manipulators in the real and virtual scenes.
4) The comparison of the proposed method and other SOTA methods [5], [8].

A. Numerical Example

In this section, a simple example with 2-dimension states is given to demonstrate the CBF in (14). We test our result on a linear system, with a polytopic obstacle defined on the state space. The system is given by

$$\dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u. \tag{21}$$

The polytopic obstacle is nonconvex with six vertices $v_1 = [-0.5; -1]$, $v_2 = [1.5; -0.5]$, $v_3 = [1; 0.5]$, $v_4 = [0.5; 0.5]$, $v_5 = [-0.5; 1]$, $v_6 = [1; -0.5]$. The system has a control limitation of $||u||_2^2 \leq 22$. We then construct the CBF defined in (14) using CVX for MATLAB, successfully construct a CBF $b(x)$ with $\boldsymbol{H}_b = \begin{bmatrix} -0.9964 & 0.0182 \\ 0.0182 & -0.9016 \end{bmatrix}$ and $d_b = 2.6893$. The level set $b(x) = 0$ together with the obstacle is shown in Fig. 2.

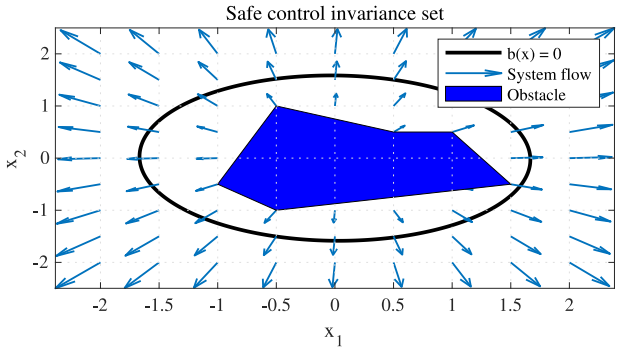

Fig. 2. Phase portrait of the control invariant set, defined by the synthesized CBF. The blue polygon represents the nonconvex obstacle on the state space, while the outside of the ellipsoid is invariant. The blue arrows indicate the vector field, which points outward on the boundary of the invariant set.

B. Control Synthesis

We preconstruct a trajectory as a series of way-points by planning methods for the manipulators, but it is not necessarily safe. $k(\boldsymbol{x})$ in (SCB-QP) is computed with a P controller to the next way-point. We set a way-point switch mechanism to avoid the manipulator getting stuck. As pointed out in [8, Proposition 1], (SCB-QP) can guarantee safety with the kinematic model of robotic manipulators when using an exponential stable low-level velocity tracking controller for Kinova Gen3. This means the tracking error converges exponentially. Specifically, the states $\boldsymbol{x}$ are the configuration of each joint, and the inputs $\boldsymbol{u}$ are the velocity of each joint.

The collision-free behavior is enforced for the kinematic model of the manipulator by constructing the CBF b online through Algorithm 1, and solving (SCB-QP).

C. Implementation

We implement our method for manipulator motion control in obstacle-cluttered environments to validate the efficacy. The manipulator, obstacles, and objects are a series of fine-shaped meshes (0.02-mm tolerance). Furthermore, the manipulator is described in a unified robot description format (URDF) in the simulation environment with *Robotics Toolbox* in MATLAB. Two real experimental scenarios are given to test these methods. We aim to use a Kinova Gen 3 robotic manipulator to grasp an object behind a board on a shelf. Two virtual scenarios are also given to test these methods in a much more complex environment. Our method first requires determining the number of samples needed to construct (SCSOS-CBF) by Algorithm 2. Then, (SCSOS-CBF) is solved with *Sedumi* and *SOSTOOL* [35] based on Algorithm 1. Finally, (SCB-QP) is solved, and the command is sent to the manipulator. All these procedures are carried out on a computer with an Intel Core i9-9980XE CPU, 3.00-GHz processor, and 64-GB RAM.

1) Computation Time of the CBF Construction Method: Each scene is tested ten times with different numbers of samples and tolerance settings of the solver. Consequently, the theoretical violation probability is quite different, but *all the tests are successfully operated with safety in practice.* The theoretical risk gives the lower bound of the violation probability. According to Theorem 2, the theoretical violation probability decreases as the sampling number grows. When aiming to control the manipulator at a low-violation probability, more SDF samples are needed to use for CBF construction. We test the proposed CBF construction method under different numbers of samples to show the computation time. The Ziggurat method generates a uniform distribution for sampling in practice [36]. In Algorithm 2, we set the $\beta = 0.05$, ϵ from 0.5 to 0.05, and the results are shown in Table II. Moreover, we repeated the experiments ten times with a tolerance of 10^{-5} to show the relationship between different violation probabilities and computation time in Fig. 3. Fig. 3 shows the maximum and minimum computation times. The computation speed is stable, and the computation time is tens of milliseconds, indicating that the proposed method can be used for online motion control. The computation time increases as the violation probability decreases. To control the manipulator at a high frequency, we refer users to set a relatively high-violation probability and obtain a short CBF construction computation time. From the experimental validation, the time complexity of our method scales well with the tolerance e_t. When the tolerance decreases for several magnitudes, the computation time increases moderately. The main influential factor is the number of samples. This is because the number of constraints in the proposed scenario program (16) increases linearly with the number of samples.

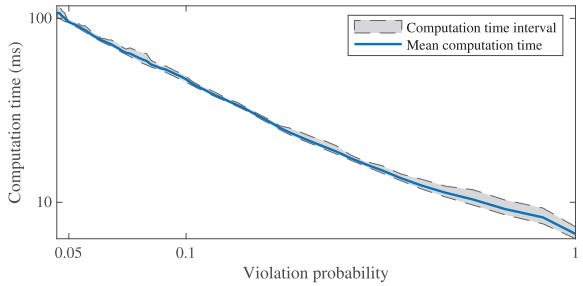

Fig. 3. Computation time of the online CBF construction under different violation probability. The tolerance of the solver is set to 10^{-5}.

TABLE II
THEORETICAL VIOLATION PROBABILITY AND COMPUTATION TIME OF THE ONLINE CBF CONSTRUCTION METHOD

Risk	Samples	Computation time (ms) with tolerance e_t		
		$e_t = 10^{-1}$	$e_t = 10^{-5}$	$e_t = 10^{-9}$
0.5	138	10.5	10.6	10.9
0.4	177	13.1	13.6	13.8
0.3	240	16.2	16.3	17.0
0.2	368	22.1	22.3	25.8
0.1	750	45.1	46.4	46.8
0.05	1513	94.1	94.4	94.9

2) Implementations of the Safety-Critical Motion Control: The primary obstacle of concern in the real scene is the shelf made of eight boards and one block on the desk. The object is a solid glue behind the block on the shelf. This scene stands for a typical application where the manipulator tends to grasp something in a complicated and cluttered indoor environment. For example, use a manipulator to grab and unplug a charger behind an LCD monitor. When performing such tasks, the

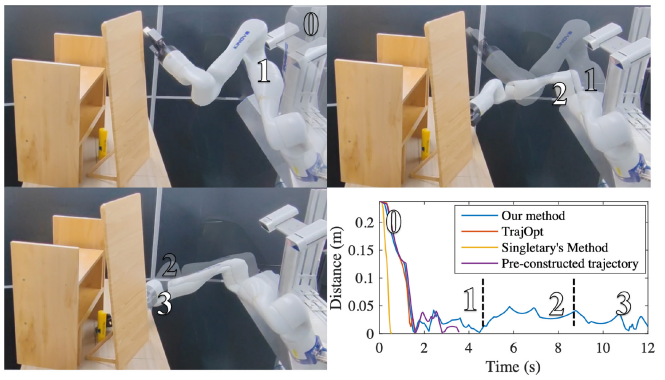

Fig. 4. Experiment implemented on the Kinova Gen3. The manipulator reaches an object on the left side. The bold Arabic numerals stand for different phases of the obstacle avoidance task. 0: the initial pose; 1: move toward the object; 2: avoid the board; and 3: reach the object. The smallest distance between the manipulator and the obstacle is about one millimeter.

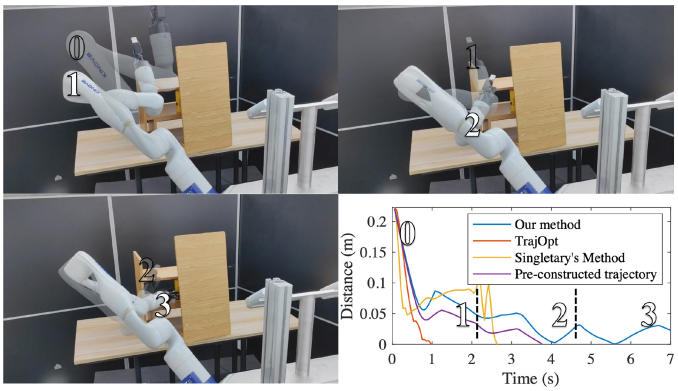

Fig. 5. Experiment implemented on the Kinova Gen3. The manipulator reached an object on the right side. 0: the initial pose; 1: move toward the board; 2: move across the board; and 3: reach the object. The smallest distance between the manipulator and the obstacle is about 2 mm.

clearance between the manipulator and the obstacles is less than a few millimeters. The number of samples is set to 400, and the safe probability is at least 81%. Our methods can achieve such tasks since the SDF is not linearized, thus preserving the volume of the feasible space for obtaining the controller. The online CBF construction further guarantees safety for model (1). Figs. 4 and 5 show the motions and distances throughout the experiment.

For comparison purposes, we also conducted the test based on CBF [8], and TrajOpt [5]. In Fig. 5, all the methods are capable of driving the manipulator to the object without collisions at the beginning. TrajOpt first collides with the obstacles since the multiple constraints in the space cause infeasibility in solving the optimization problem. The method in [8] did not achieve the goal either since the linearization of SDF caused the infeasibility problem. Moreover, the manipulator is driven based on the preconstructed trajectory. The distance is given to show that the preconstructed trajectory is not necessarily safe in our study.

We then design a much more complex environment in the virtual scene to further demonstrate the performance limit of the proposed method. The obstacles of concern are eight balls, four pillars, and one giant board. The object is on the other side of the board. The manipulator must pass through the window at

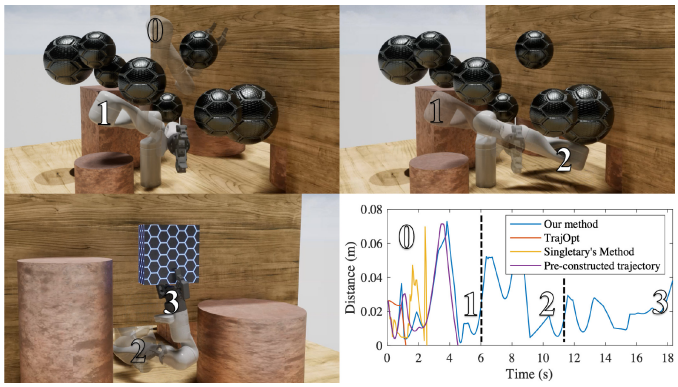

Fig. 6. Experiment implemented on the Kinova Gen3 in a virtual environment with static obstacles. 0: the initial pose; 1: move downward; 2: move through the window; and 3: reach the object. The smallest distance between the manipulator and the obstacle is about 4 mm.

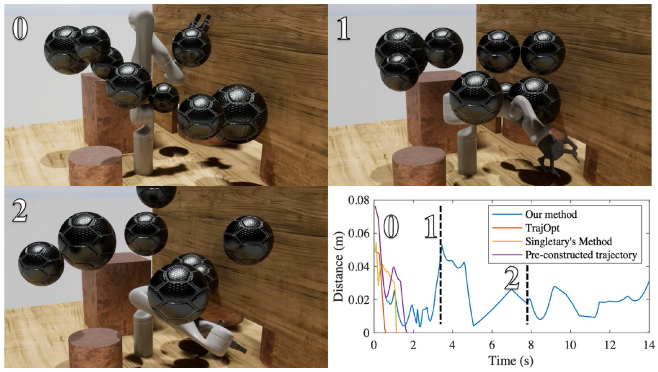

Fig. 7. Experiment implemented on the Kinova Gen3 in a virtual environment with moving obstacles. The SDF is sampled online.

the bottom of the board while avoiding collision. Our method can achieve this task with a clearance of about 4 mm. Fig. 6 shows the virtual motion and the distance throughout the trajectory. We continued to simulate the virtual scene where the eight balls were moving. Fig. 7 shows the virtual test with moving obstacles and the distance throughout the motion.[1] Similarly, the other methods cannot reach the goal, indicating that our method can perform better in a cluttered environment.

As for the computation time of solving the optimal control problem, the relaxation in (SCSOS-CBF) guarantees our method has a real-time ability. As for the average QP computation time, it is 12.41 ms. The average computation time for QP in CBF is 12.34 ms, similar to our result. The average computation time for TrajOpt is 270 ms. The average computation time for our method is 39.31 ms. Moreover, an accelerated method that shortens the average computation time bypasses the CBF construction procedure when all samples are above zero. Specifically, when $\forall\, \boldsymbol{x}^{(i)} \in \bar{\boldsymbol{X}},\ h(\boldsymbol{x}^{(i)}) \geq 0$, directly using $\boldsymbol{u}^*(\boldsymbol{x}) = k(\boldsymbol{x})$ can avoid solving the optimization problems.

In conclusion, the method in [8] and TrajOpt cannot be successful in all tests, especially when the clearance is within millimeters. Our method is better and more suitable for online safety-critical motion control in a cluttered environment.

[1]The video can be found at https://www.youtube.com/watch?v=PJzaJ0QG7Hg.

Safety standards in robotics, such as ISO 10218-1, 10218-2, 13854, and 13857, provide minimum gap requirements to avoid crushing human bodies for industrial manipulators. Our study proposed a method for manipulators to avoid collision in a cluttered environment where the potential distance is within millimeters. The simulations and experiments showed the effectiveness of the proposed method. Therefore, our method can meet the requirements of the safety standards in robotics [37] when dealing with obstacle-avoidance tasks. The proposed method can also improve safety and transparency in the IEEE 7007 standard [38]. The proposed can be used as a filter in the action modules of the IEEE 7007 unified modeling language model. The constructed CBF can be shared with other manipulators in a multiagent scenario to meet the transparency requirements [39].

V. Conclusion

In this article, an online CBF construction method is proposed. The method tails the characteristic of the system model to ensure safety under the merit of control invariance. Our method delivers less conservative results with a probability guarantee by sampling data and incorporating the samplings as scenarios in the optimization problem. We also validate the efficacy of the proposed algorithm on industrial robotic manipulators to perform several safety-critical control tasks.

Despite the advantages of the proposed method demonstrated above, it also has some limitations that need to be studied.

1) The computation time of CBF construction is extended when the number of samples is high. How to efficiently solve the SDP of (17) is an interesting problem to be studied.
2) The probability given in Theorem 2 shows a lower-bound result.

Thus, the samples used for constructing CBF are too much when given an unsafe risk. How to tighten the results in Theorem 2 is another problem of interest.

References

[1] X. Guo, H. Zhang, J. Sun, and Y. Zhou, "Fixed-time fuzzy adaptive control of manipulator systems under multiple constraints: A modified dynamic surface control approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 4, pp. 2522–2532, Apr. 2023.

[2] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, "Manipulator motion planning for part pickup and transport operations from a moving base," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 1, pp. 191–206, Jan. 2022.

[3] R. Wang, X. Zhang, Y. Fang, and B. Li, "Virtual-goal-guided RRT for visual servoing of mobile robots with FOV constraint," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2073–2083, Apr. 2022.

[4] S.-H. Jung, G. Lee, and D. Han, "Methods and tools to construct a global indoor positioning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 906–919, Jun. 2018.

[5] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.

[6] D. Malyuta et al., "Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently," *IEEE Control Syst. Mag.*, vol. 42, no. 5, pp. 40–113, Oct. 2022.

[7] S. Ruan, K. L. Poblete, H. Wu, Q. Ma, and G. S. Chirikjian, "Efficient path planning in narrow passages for robots with ellipsoidal components," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 110–127, Feb. 2023.

[8] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10954–10961, Oct. 2022.

[9] H. Ding, G. Reißig, D. Groß, and O. Stursberg, "Mixed-integer programming for optimal path planning of robotic manipulators," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2011, pp. 133–138.

[10] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 972–983, May 2021.

[11] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[12] L. Kong, J. Reis, W. He, and C. Silvestre, "Experimental validation of a robust prescribed performance nonlinear controller for an unmanned aerial vehicle with unknown mass," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 1, pp. 301–312, Feb. 2024.

[13] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988.

[14] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3D game objects," in *Proc. Game Develop. Conf.*, 2001, pp. 1–17.

[15] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3665–3670, Jul. 2017.

[16] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf. (ECC)*, 2019, pp. 3420–3431.

[17] M. Korda, D. Henrion, and C. N. Jones, "Convex computation of the maximum controlled invariant set for polynomial control systems," *SIAM J. Control Optim.*, vol. 52, no. 5, pp. 2944–2969, 2014.

[18] L. Kong, J. Reis, W. He, and C. Silvestre, "Comprehensive nonlinear control strategy for VTOL-UAVs with windowed output constraints," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2673–2684, Nov. 2023.

[19] Y. Zheng and K. Yamane, "Generalized distance between compact convex sets: Algorithms and applications," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 988–1003, Aug. 2015.

[20] Y. Zheng and K. Hang, "Calculating the support function of complex continuous surfaces with applications to minimum distance computation and optimal grasp planning," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1004–1021, Aug. 2020.

[21] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 1366–1373.

[22] L. Han, F. Gao, B. Zhou, and S. Shen, "FIESTA: Fast incremental Euclidean distance fields for online motion planning of aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 4423–4430.

[23] J. Pankert and M. Hutter, "Perceptive model predictive control for continuous mobile manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6177–6184, Oct. 2020.

[24] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022, pp. 286–292.

[25] M. A. Duarte-Mermoud, N. Aguila-Camacho, J. A. Gallegos, and R. Castro-Linares, "Using general quadratic Lyapunov functions to prove Lyapunov uniform stability for fractional order systems," *Commun. Nonlin. Sci. Numer. Simul.*, vol. 22, nos. 1–3, pp. 650–659, 2015.

[26] K. Derinkuyu and M. Ç. Pınar, "On the S-procedure and some variants," *Math. Methods Oper. Res.*, vol. 64, no. 1, pp. 55–77, 2006.

[27] J. Anderson and A. Papachristodoulou, "Advances in computational Lyapunov analysis using sum-of-squares programming," *Discrete Continuous Dyn. Syst. Ser. B*, vol. 20, no. 8, pp. 2361–2381.

[28] H. Wang, K. Margellos, and A. Papachristodoulou, "Safe controlled invariance for linear systems using sum-of-squares programming," 2022, *arXiv:2207.00321*.

[29] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 625–653, 1999.

[30] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1211–1230, 2008.
[31] S. Garatti and M. C. Campi, "Risk and complexity in scenario optimization," *Math. Program.*, vol. 191, pp. 243–279, Jan. 2022.
[32] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, *Path Planning and Trajectory Planning Algorithms: A General Overview*, vol. 29. Cham, Switzerland: Springer, Mar. 2016, pp. 3–27.
[33] T. Rybus, "Point-to-point motion planning of a free-floating space manipulator using the rapidly-exploring random trees (RRT) method," *Robotica*, vol. 38, no. 6, pp. 957–982, 2020.
[34] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
[35] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. Parrilo, "SOSTOOLS version 4.00 sum of squares optimization toolbox for MATLAB," 2013, *arXiv:1310.4716*.
[36] C. B. Moler, *Numerical Computing with MATLAB*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2004.
[37] T. Jacobs, J. Veneman, G. S. Virk, and T. Haidegger, "The flourishing landscape of robot Standardization [industrial activities]," *IEEE Robot. Autom. Mag.*, vol. 25, no. 1, pp. 8–15, Mar. 2018.
[38] E. Prestes et al., "The first global ontological standard for ethically driven robotics and automation systems [standards]," *IEEE Robot. Autom. Mag.*, vol. 28, no. 4, pp. 120–124, Dec. 2021.
[39] H. Wang, A. Papachristodoulou, and K. Margellos, "Distributed control design and safety verification for multi-agent systems," 2023, *arXiv:2303.12610*.

Xuda Ding (Student Member, IEEE) received B.E. and M.E. degrees in automatic control from the School of Automation, North China Electric Power University, Beijing, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China.

His research interests include data-driven modeling, safety-critical control, and digital twin system in industrial applications.

Mr. Ding is a member of the Intelligent Wireless Networking and Cooperative Control Group.

Han Wang received the B.S. degree in information security from Shanghai Jiao Tong University, Shanghai, China, in 2020. He is currently pursuing the Ph.D. degree in engineering science with the Department of Engineering Science, University of Oxford, Oxford, U.K.

His research interests include safety verification and safe control design.

Mr. Wang is a member of Control Group.

Yi Ren received the B.Sc. degree in thermal energy and power engineering from the Harbin Institute of Technology, Harbin, China, in 2011, and the M.Sc. and Ph.D. degrees in mechatronics engineering from the State Key Laboratory of Robotics and System, Harbin Institute of Technology in 2013 and 2017, respectively.

Since 2022, he has been a Technical Expert with the Advanced Manufacturing Lab, Huawei Technologies, Shenzhen, China. His research interests include robotics, nonlinear control, distributed control, perception, and natural language processing with application to multirobot cooperation and manipulation.

Yu Zheng (Senior Member, IEEE) received the B.Eng. degree in mechanical engineering and computer science from Shanghai Jiao Tong University, Shanghai, China, in 2001, the M.S. degree in computer science from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2011, the first Ph.D. degree in mechatronics from Shanghai Jiao Tong University in 2007, and the second Ph.D. degree in computer science from the University of North Carolina at Chapel Hill in 2014.

He was with Tencent Robotics X Laboratory, Shenzhen, China, in September 2018, and is currently a Principal Research Scientist and the Team Lead of the Control Center. His research interests include multicontact/multibody robotic systems, robotic grasping and manipulation, legged robots, and various algorithms for robotics.

Dr. Zheng serves as an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS.

Cailian Chen (Member, IEEE) received the B.Eng. and M.Eng. degrees in automatic control from Yanshan University, Qinhuangdao, China, in 2000 and 2002, respectively, and the Ph.D. degree in control and systems from The City University of Hong Kong, Hong Kong, in 2006.

She has been with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, since 2008. She is currently a Distinguished Professor. She has been actively involved in various professional services. Her research interests include industrial wireless networks and computational intelligence, and Internet of Vehicles.

Prof. Chen serves as a Deputy Editor for *National Science Open* and an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *IET Cyber-Physical Systems: Theory and Applications*, and *Peer-to-Peer Networking and Applications* (Springer). She also served as a Guest Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, a TPC Chair for ISAS'19, a Symposium TPC Co-Chair for IEEE Globecom 2016, a Track Co-Chair for VTC2016-fall and VTC2020-fall, and a Workshop Co-Chair for WiOpt'18.

Jianping He (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2013.

He had been a Research Fellow with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, from December 2013 to March 2017. He is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His research interests mainly include the distributed learning, control, and optimization, and security and privacy in network systems.

Dr. He serves as an Associate Editor for IEEE TRANSACTIONS CONTROL OF NETWORK SYSTEMS, *IEEE Open Journal of Vehicular Technology*, and *KSII Transactions Internet and Information Systems*. He was also a Guest Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL and *International Journal of Robust and Non-linear Control*.