

Safety-Critical Optimal Control for Robotic Manipulators in A Cluttered Environment

Xuda Ding¹, Han Wang², Yi Ren³, Kostas Margellos², Yu Zheng³, Jianping He¹, Cailian Chen¹

Abstract—Designing safety-critical control for robotic manipulator is challenging, especially in a cluttered environment. 1) The actual trajectory might deviate from the planned one due to the complex collision environments and non-trivial dynamics, leads to collision; 2) The feasible space for the manipulator is hard to obtain since the explicit distance functions between collision meshes is unknown. This paper proposes a data-driven control barrier function (CBF) construction method, which extracts CBF from distance samples. The CBF guarantees the safety for considering the system dynamic. The data-driven method approximates the distance function and determines the safe set. Then, the CBF is synthesized based on safe set by a scenario-based sum-of-square (SOS) program. Unlike the existing linearization-based approach, our method enlarges the volume of the feasible space for planning. The control law is obtained by solving a quadratic programming problem with CBF in real-time, which works as a safety filter for the desired planning-based controller. Moreover, the proposed method guarantees safety with the proved probabilistic result. The proposed method is validated on a 7-DOF manipulator in both real and virtual cluttered environments. The experiments show that the manipulator is able to actuate tasks where the clearance between obstacles is in millimeters.

I. INTRODUCTION

Safety-critical motion planning is fundamental for applications of robotic manipulators since manipulators need to be driven to a specified goal without collisions [1], [2]. The whole body must have no collisions with obstacles and itself. Due to the complex dynamics and the cluttered environments where the potential clearance between the manipulator and the obstacles is less than a few millimetres, motion planning is challenging. In the past decades, path planning methods combined with tracking control have been proposed to generate a safe path for manipulators based on the environment map. Encouraged by the efficient applications of graph search methods in path planning, Rapidly-Exploring Random Trees (RRT) [2] and Probabilistic Road Map (PRM) [3] are proposed to deal with planning problems with high-dimensional configuration spaces. The randomized methods are probabilistically complete, which means they may take a long time to achieve asymptotic optimality. To deal with the optimality problem, the non-linear optimal algorithms such

This work is supported by the CIE-Tencent Robotics X Rhino-Bird Focused Research Program.

¹: the Dept. of Automation, Shanghai Jiao Tong University, Shanghai, China. email: {dingxuda, jphe, cailianchen}@sjtu.edu.cn.

²: the Dept. of Engineering Science, University of Oxford, Oxford, United Kingdom. E-mails: {han.wang, kostas.margellos}@eng.ox.ac.uk

³: the Tencent Robotics X Lab, Shenzhen, China. email: {evanyren, petezheng}@tencent.com

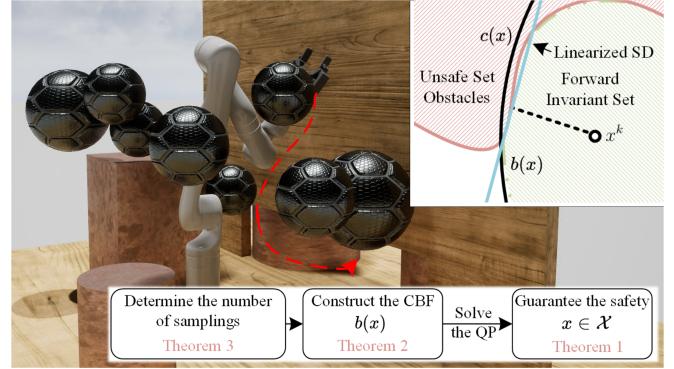


Fig. 1: The proposed safety-critical optimal control framework for the manipulator. The linearized SDF at x^k overlooks some feasible space. Our approach uses more sample of the SDF to construct the control invariant set, which enlarge the volume of the feasible space.

as TrajOpt [4], CHOMP [5] and OMPL [6] are used to plan joint trajectories offline. Offline planning requires a global map of the obstacles to making plans. However, when the map is unknown, the solution of the offline methods would fail. Further, since the actual trajectory might deviate from the planned one [7], safety cannot be guaranteed.

System-dynamic-based optimization methods are proposed to deal with motion 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 planning can satisfy the collision-free requirement. Mixed Integer Linear Problem (MILP) and Mixed Integer Quadratic Problem (MIQP) are proposed to generate the motion plan by sampling and calculating the points on the manipulator's surface to the obstacles [8], [9]. Since the number of integer variables constraints is large, MILP/MIQP takes seconds to minutes to solve. Without integer variables, the QP approach is used for motion plans whose computing speed is much faster than MILP/MIQP [10], [11]. In recent, with the revisiting of the controlled invariant set and Control Barrier Function (CBF) in control theory, QP with CBF-based constraints approach is used for safe-critical motion plan and enforce the safety [7], [12]. However, it is still a challenging problem for QP to construct constraint functions concerning collision even with a complete environment map since the manipulator's configuration space does not match the obstacles' space. The mainstream method is a sampling-based one, which is proposed to obtain the Signed Distance Function (SDF) between two objects considering the meshes [4], [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. Directly using SDF in QP can lead to a local minimum and disobey the CBF-based optimization requirement of continuity [7]. Linearization of the SDF is used to solve this problem, and then the constraints are constructed [4], [7], [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 [4]. Furthermore, the linearization of SDF or SDF gives a determination for safety, but they do not necessarily equal CBF since neither guarantees control invariance.

This letter aims to design a real-time optimal control method in a cluttered environment and enforces safety based on the CBF. Specifically, the CBF is extracted based on the safe set where collision is free. The extraction of the CBF considers the manipulator's dynamics and guarantees control invariance and safety. The safe set is determined from online SDF sampling data. The SDF is not linearized, so the safe set space and the feasible space volume are enlarged for the optimization problem of getting a feasible result in a cluttered environment. Some critical challenges need to be solved for the safety-critical motion plan of manipulators. i) Since the explicit SDF cannot be determined, the safe set is unknown. ii) How to extract a candidate CBF based on a safe set since the SDF may not guarantee the forward invariant property. iii) The number of SDF samples needed for estimating the safe set and extracting the CBF, the probability of safety still need to be determined. Inspired by the scenario approach and CBF construction [15], [16], we found it is possible to solve these challenges. Our approach first formulates the relationship between the safe set based on SDF, considering the collisions with obstacles and the manipulator itself. Then, 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 shows the proposed safety-critical optimal control framework for manipulators in a cluttered environment.

The main contributions of this letter are

- A data-driven CBF construction methodology is proposed. By solving a scenario-based sum-of-square (SOS) program, CBF is efficiently synthesized according to the dynamics model. The construction does not rely on the linearized SDF, thus enlarging the feasible space for planning.
- Unsafety probability is characterized based on the number of samples and support constraints. The probability guides the sampling number used for CBF construction and safety-critical control.
- The proposed approach is implemented on a full-scale manipulator (Kinova Gen3) in a obstacles cluttered

environment. The speed and efficacy of this method are extensively explored in real-world environments, and the method has been shown to increase planning speed and reliability dramatically.

The rest of this paper is organized as follows. Section II formulates the optimal control problem with CBF, and introduces the relationship between safety and control invariant set and the SDF. The data-driven CBF extraction method and the probability guarantee for safety are given in Section III. Simulation and implementation are shown in Section IV. Section V concludes the paper.

II. PRELIMINARIES

A. System Dynamics and Objective

Consider a dynamic model of robotic manipulator

$$\dot{x} = f(x, u), \quad (1)$$

where state $x(t) \in \mathcal{X} \subset \mathbb{R}^n$ and input $u(t) \in \mathcal{U} \subset \mathbb{R}^m$, \mathcal{U} is a compact set and $f(\cdot, \cdot)$ is Lipschitz continuous. Our goal is to find a control policy which allows the system to navigate the manipulator \mathcal{A} from initial state x_0 to final state x_T while optimizing a objective function and avoiding collision with obstacles $\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_M \subset \mathbb{R}^3$, where $M \in \mathbb{Z}^+$.

To guarantee safety, we must ensure that the state of the manipulator \mathcal{A} is kept within a set \mathcal{B} where the manipulator is collision-free. Such set \mathcal{B} is a *Controlled Invariant Set*.

Definition 1 (Controlled Invariant Set). *A set \mathcal{B} is called a controlled invariant with respect to (1) if for every $x_0 \in \mathcal{B}$, there exists an input $u(t)$ such that $x(t) \in \mathcal{B}$ for time $t \in [0, t_{\max}]$. When $t_{\max} = \infty$, we say $f(\cdot, \cdot)$ is forward complete.*

Controlled Invariant Set guarantees the states of the controlled object flow in the region with respect to the system dynamics [17]. Roughly speaking, the states must be in the set \mathcal{B} and never leave. The CBF is denoted by a \mathbb{C}^1 function $b(x) : \mathbb{R}^n \rightarrow \mathbb{R}$. The corresponding controlled invariant set \mathcal{B} is the zero super-level set of $b(x)$. The CBF $b(x)$ can be incorporated into a QP to synthesize point-wise optimal and safe control law [7], [12].

$$\begin{aligned} u^*(x) = \arg \min_u & \frac{1}{2} \|u - u_{\text{des}}(x)\|_2^2 \\ \text{subject to } & \frac{\partial b(x)}{\partial x} f(x, u) + \lambda b(x) \geq 0, \end{aligned} \quad (\text{SCB-QP})$$

where $u_{\text{des}}(x)$ is a desired input. $u^*(x)$ is the safe input, and λ is a relaxation coefficient to reduce restriction in QP. This QP can be solved in real-time for nonlinear systems. How to construct a $b(x)$ and make sure the *controlled invariant set* is a subset of the safe set are critical problems addressed in this paper.

B. Safety and SDF

Here we define the *safe set* to determine the safety of the *controlled invariant set*.

Definition 2 (Safe Set). *To guarantee the manipulator is safe without collision, the states $\{x(t)\}$ should always in the*

collision-free set. Suppose there exists a series of functions $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$\begin{aligned}\mathcal{X} &= \cap_{i=1}^{\mathcal{I}} \{x \in \mathbb{R}^n | h_i(x) \geq 0\}, \\ \text{Int}(\mathcal{X}) &= \cap_{i=1}^{\mathcal{I}} \{x \in \mathbb{R}^n | h_i(x) > 0\}, \\ \bar{\mathcal{X}} &= \cup_{i=1}^{\mathcal{I}} \{x \in \mathbb{R}^n | h_i(x) < 0\},\end{aligned}\quad (2)$$

where $\text{Int}(\mathcal{X})$ and $\bar{\mathcal{X}}$ are the interior and complementary set of \mathcal{X} , respectively. The sequence of functions $h_1(x), \dots, h_{\mathcal{I}}(x)$ describe the safe set, and represent the obstacles which are indexed by $1, \dots, \mathcal{I}$.

Our goal is to design controller u to guarantee the manipulator within the safe set \mathcal{X} .

The existence of such controller u is promised by the following lemma.

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

Remark. *It should be noted that the function $h(x)$ used for establishing a safe set is not necessarily the same as the CBF $b(x)$, for the safe set constructed based on the obstacles may not guarantee the invariance property with respect to dynamics (1).*

The meshes of the obstacles and the manipulator are considered to construct the safe set \mathcal{X} and $h_i(x)$. When the global information about the obstacles is known, the SDF $\text{sd}(\mathcal{A}, \mathcal{O}_i)$ between manipulator \mathcal{A} and the obstacle \mathcal{O}_i can be obtained based on a popular way [4], [13], [14]

$$\text{dist}(\mathcal{A}, \mathcal{O}_i) := \min_d \{\|d\| : (\mathcal{A} + d) \cap \mathcal{O}_i \neq \emptyset\}, \quad (3a)$$

$$\text{pen}(\mathcal{A}, \mathcal{O}_i) := \min_d \{\|d\| : (\mathcal{A} + d) \cap \mathcal{O}_i = \emptyset\}, \quad (3b)$$

$$\text{sd}(\mathcal{A}, \mathcal{O}_i) := \text{dist}(\mathcal{A}, \mathcal{O}_i) - \text{pen}(\mathcal{A}, \mathcal{O}_i), \quad (3c)$$

where d is the translation of the manipulator. When global information cannot be obtained, the obstacles need to be sampled and constructed. *Voxblox* [18] and *FIESTA* [19] can be used to incrementally construct the Euclidian Signed Distance Fields (ESDF) of the obstacles and the signed distance $\text{sd}(\mathcal{A}, \mathcal{O}_i)$ is obtained. Then, two points $p_{\mathcal{A}}, p_{\mathcal{O}_i}$ on the manipulator and the obstacle coressponding to (3) are obtained. Note that $p_{\mathcal{A}}, p_{\mathcal{O}_i}$ are points expressed in their own local coordinates, and they are needed to be transferred into world coordinates. Two points in the world coordinates are $F_{\mathcal{A}}^w(x)p_{\mathcal{A}} \in \mathbb{R}^3$ and $F_{\mathcal{O}_i}^w p_{\mathcal{O}_i} \in \mathbb{R}^3$. The function F^w is forward kinematics, which gives the pose of the manipulator and the obstacles in the world frame. The SDF is nonsmooth, and its explicit form is hard to obtain. Linearization of SDF is used for convex formulation in [4], [7], [11]. In this work, we propose to directly extract the SDF based on samplings to acquire less conservativeness.

Overall, we aim to extract a safe-critical CBF based on samplings of SDF to guarantee collision-free manipulation while considering the system dynamic intrinsically.

III. EXTRACT CBF FROM SDF SAMPLING CONSIDERING DYNAMIC MODEL

This section shows the main results of our CBF construction method based on SDF sampling. Our approach is data-driven since the explicit form of SDF cannot be obtained. The CBF is constructed based on the SDF samplings. This section is organized in three parts. In Section III-A we formulate the *safe set* based on SDF. In Section III-B, CBF construction method with SOS program is proposed. Based on the samples, a data-driven CBF construction method is given with the probabilistic guarantee for safety in Section III-C.

A. Safe Set Construction based on SDF

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

$$\text{sd}_{\text{ot}}(x) = \max_{\|\hat{n}_{\text{ot}}\|_2=1} \min_{\substack{p_{\mathcal{A}} \in \mathcal{A} \\ p_{\mathcal{O}_i} \in \mathcal{O}_i}} \hat{n}_{\text{ot}} \cdot (F_{\mathcal{A}}^w(x)p_{\mathcal{A}} - F_{\mathcal{O}_i}^w(x)p_{\mathcal{O}_i}). \quad (4)$$

where \hat{n}_{ot} is the direction of the minimal translation d in (3). $\text{sd}_{\text{ot}}(x)$ 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 scenarios considered in the last subsection, another possible collision scenario happens for different components of the manipulator. For this case, the inner-SDF is defined by:

$$\text{sd}_{\text{in}}(x) = \max_{\|\hat{n}_{\text{in}}\|_2=1} \min_{\substack{p_{\mathcal{A}} \in \mathcal{A} \\ p'_{\mathcal{A}} \in \mathcal{A}}} \hat{n}_{\text{in}} \cdot (F_{\mathcal{A}}^w(x)p_{\mathcal{A}} - F_{\mathcal{A}}^w(x)p'_{\mathcal{A}}), \quad (5)$$

where $p_{\mathcal{A}}$ and $p'_{\mathcal{A}}$ are different points on the different joints of the manipulator. Then, the safe set of the manipulator is constructed as the following result.

Theorem 1. *Given the outer-SDF (4) and inner-SDF (5), safe set \mathcal{X} of the manipulator is:*

$$\mathcal{X} := \{x | \text{sd}_{\text{ov}}(x) \geq 0\}. \quad (6)$$

where $\text{sd}_{\text{ov}}(x) = \min\{\text{sd}_{\text{ot}}(x), \text{sd}_{\text{in}}(x)\}$ is the overall-SDF derived from (4) and (5).

Proof. The composition relationship in $\text{sd}_{\text{ov}}(x)$ is captured by a \wedge quantifier. If for a x , the manipulator is both inner-collision free and outer-collision free, then $\text{sd}_{\text{in}}(x) \geq 0 \wedge \text{sd}_{\text{ot}}(x) \geq 0$, which is equivalent to $\text{sd}_{\text{ov}}(x) \geq 0$. When $\text{sd}_{\text{ov}}(x) = 0$, the manipulator is at the boundary of collision; When $\text{sd}_{\text{ov}}(x) > 0$, the manipulator is away from collision; When $\text{sd}_{\text{ov}}(x) < 0$, the collision happens. Therefore, the set \mathcal{X} is a safe set. \square

B. CBF Construction Considering Safe Set and Dynamic Model

In most of the existing literature, the signed distance is directly used or linearized then used as a CBF in safety-critical controller design problems [7], [11], [20]. However, the safe set \mathcal{X} defined by the zero-super level set of distance function is not necessary to be a candidate CBF, as it has no controlled invariance property. Actually, the whole construction procedure in Section III-A utilizes no formal knowledge of the manipulator's dynamic model (1). In this subsection, we show how to synthesize a CBF $b(x)$ from the safe set \mathcal{X} based on overall-SDF $\text{sd}_{\text{ov}}(x)$.

Lemma 2. *For manipulator \mathcal{A} with dynamics (1), and overall-SDF $\text{sd}_{\text{ov}}(x)$, $b(x)$ is a candidate CBF if*

$$b(x) \leq \text{sd}_{\text{ov}}(x), \quad (7a)$$

$$\forall x \in \partial \mathcal{B}, \exists u \in \mathcal{U}, \quad \frac{\partial b(x)}{\partial x} f(x, u) \geq 0. \quad (7b)$$

Proof. The right hand side inequality in (7a) stands for that the CBF $b(x)$ should be a lower-envelope of the overall-distance function $\text{sd}_{\text{ov}}(x)$. Under this property, if $b(x) \geq 0$ for any x on the motion trajectory then $\text{sd}_{\text{ov}}(x) \geq 0$, and safety can be guaranteed. The second Condition (7b) is a standard controlled invariance condition, details about this condition can be found in [21]. \square

Without an additional objective, the construction of $b(x)$ leads to the following feasibility optimization problem.

$$\text{find } b(x), \text{s.t. (7).} \quad (8)$$

Although the conditions are algebraic structured elegant, it is still very hard to construct a CBF $b(x)$ by solving (8). The challenges here are threefold: i) Conditions (7a) and (7b) should hold for infinite x , which renders (8) to be an infinitely-constrained optimisation problem; ii) $\text{sd}_{\text{ov}}(x)$ has only an implicit form since it is a composition of solutions to two optimisation problems; iii) The constraint (7b) is especially nonconvex due to the existence of control input u . All these three challenges will be tackled in this section, but we want to point out here that the computational complexity of our method is still high. The reason is that the explicit $\text{sd}_{\text{ov}}(x)$ cannot be obtained. Estimation of $\text{sd}_{\text{ov}}(x)$ based on samplings costs time. We will give a detailed discussion after showing the CBF construction algorithm 1. Then, we will determine the set \mathcal{C} where the samples are obtained to construct CBF $b(x)$.

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

$$\|\delta x\|_{x=x^k} = \max_{u \in \mathcal{U}} \|f(x^k, u) - x^k\|^2. \quad (9)$$

The set \mathcal{C} at state x^k is then defined by a ball $\mathcal{B}\mathcal{A}(x^k, \|\delta x\|_{x=x^k})$ centered on x^k , with radius $\|\delta x\|_{x=x^k}$.

The reason why we use a high dimensional ball but not the exact reachable region, i.e. $\bigcup_{u \in \mathcal{U}} f(x^k, u)$, is that $\mathcal{B}\mathcal{A}(x^k, \|\delta x\|_{x=x^k})$ has a good convexity, and is computationally cheaper. Clearly, $\bigcup_{u \in \mathcal{U}} f(x^k, u) \subseteq \mathcal{B}\mathcal{A}(x^k, \|\delta x\|_{x=x^k})$.

2) *Construct the CBF $b(x)$:* In the first part, we show how to construct a candidate CBF $b(x)$ satisfying the local nonnegativity condition $\forall x \in \mathcal{X}, 0 \leq b(x)$ in (7). $b(x)$ is parameterized by:

$$b(x) = x^\top H_b x + d_b, \quad (10)$$

where $H_b \prec 0$, and $d_b > 0$. This kind of CBF is originated from the quadratic Lyapunov function $v(x) = -x^\top H_b x$ [22], where we have $b(x) = d_b - v(x)$. We consider parameterizing the CBF $b(x)$ to be ellipsoidal because our application has high control frequency. As a result, in the state space, the Lebesgue measure of the ball $\mathcal{B}\mathcal{A}(x_k, \|\delta x\|_{x=x^k})$ would be relatively small. In such a small region, the original nonlinear system (1) can be linearized with a slight bias. 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 [16].

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

$$\begin{aligned} & \text{find } b(x), \sigma_1, \\ & \text{subject to } -b(x) + \sigma_1 c(x) \in \Sigma[x], \end{aligned} \quad (\text{SOS-CBF})$$

where $\sigma_1 \in \Sigma[x]$ is a SOS multiplier. We recall that the set \mathcal{C} is restricted to be a ball centered at x^k , i.e. $\mathcal{B}\mathcal{A}(x_k, \|\delta x\|_{x=x^k})$ in the first step. Therefore, $c(x) = -\|x - x^k\|^2 + \|\delta x\|_{x=x^k}^2$. It is evident that $c(x)$ is a polynomial function, thus \mathcal{C} is a semi-algebraic set. Together with the polynomial function $b(x)$ 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). Tab.I illustrates the different sets to provide a clear understanding.

TABLE I: Illustration of set \mathcal{X} , \mathcal{C} , \mathcal{B} .

Set	Denotation	Remark
\mathcal{X}	safe set	Defined by (6), is not explicitly applicable to the manipulator.
\mathcal{C}	reachable set	Defined by $\mathcal{B}\mathcal{A}(x^k, \ \delta x\ _{x=x^k})$, is the maximum reachable set of system (1) at x^k .
\mathcal{B}	controlled invariant set	Synthesized by (SCSOS-CBF), and satisfies: $\mathcal{B} \subseteq \mathcal{C}$, $\mathcal{B} \in \mathcal{X}$.

C. Data-driven CBF Construction

The following subsections show how to construct $b(x)$ satisfying the residual constraints in Condition (7) with a promising probabilistic bound. In addition to the SOS synthesis approach in the last part, we use scenario optimization to

Algorithm 1: Data-driven CBF construction

Input: the number of samples \bar{N} , current state x^k ,
the maximum ball set $\mathcal{BA}(x^k, \|\delta x\|_{x=x^k})$.
Output: CBF parameter 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 $\pi(x)$
- 3 **for** $i \leq \bar{N}$ **do**
- 4 | Compute $\text{sd}_{\text{ov}}(x^{(i)})$ for all samples in \bar{X} .
- 5 **end**
- 6 Solve the optimal problem (SCSOS-CBF).
- 7 Return H_b and d_b .

alleviate some of the constraints. The reason why SOS is not fully applicable for the remaining constraint $\forall x \in \mathcal{C}, b(x) \leq \text{sd}_{\text{ov}}(x)$ is that $\text{sd}_{\text{ov}}(x)$ is not a polynomial function in general. More precisely, there is even no explicit expression of it by hand. The constraint $\forall x \in \partial\mathcal{C}, \exists u \in \mathcal{U}, \frac{\partial b(x)}{\partial x} f(x, u) \geq 0$ is also hard to convert to SOS constraints as $f(x, u)$ may not be a polynomial, and exhibits bilinearity due to the existence of control input. Although there are lifting methods [23] and Schur relaxation methods [16] to overcome these issues, they either do not scale well with dimension or require iterative solutions. In our problem, 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. We recall that in our problem, the decision variables are the parameters in $b(x)$ and σ_1 . The state x can then be regarded as uncertain perturbations in a robust optimization framework [24]. Instead of solving the robust optimization problem for any $x \in \mathcal{C}$, we sample \bar{N} realizations of $x^{(r)}$ around x^k , termed scenarios with a probability measure π which satisfies:

$$\int_{\mathcal{C}} \pi(x) dx = 1. \quad (11)$$

Let $\bar{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(\bar{N})}\}$ be the set of sampled scenarios. These scenarios are independently and identically sampled according to π . We could construct the following scenario program:

$$\begin{aligned} & \text{find}_{u \in \mathcal{U}} b(x), u \\ & \text{subject to } b(x^{(i)}) \leq \text{sd}_{\text{ov}}(x^{(i)}), \\ & \quad \frac{\partial b(x)}{\partial x} f(x, u) + \alpha b(x) \Big|_{x=x^{(i)}} \geq 0, \quad (\text{SC-CBF}) \\ & \quad \forall x^{(i)} \in \bar{X}. \end{aligned}$$

The lower-envelope condition $\forall x \in \mathcal{C}, b(x) \leq \text{sd}_{\text{ov}}(x)$ is enforced only on the finite set of scenarios \bar{X} . The controlled invariance condition (7b) is substituted by a relaxed formulation $\forall x \in \mathcal{B}, \exists u \in \mathcal{U}, \frac{\partial b(x)}{\partial x} f(x, u) + \alpha b(x) \geq 0$, where $\alpha \in \mathbb{R}_+$. This relaxed formulation enables us to eliminate the nonconvexity introduced by the cross terms. The additional term $\alpha b(x)$ is motivated by the zero-CBF approach [12].

The final synthesis program for a data-driven CBF with SD samples is given as follows:

Theorem 2. For dynamical system (1), sampling a set of scenarios $\bar{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(\bar{N})}\} \in \mathcal{C}$, the CBF $b(x)$ at the current state x^k can be constructed by solving the following program:

$$\begin{aligned} & \max_{u \in \mathcal{U}, \sigma_1 \in \Sigma[x], H_b \prec 0, d_b > 0} d_b \\ & \text{subject to } b(x^{(i)}) \leq \text{sd}_{\text{ov}}(x^{(i)}), \\ & \quad \frac{\partial b(x)}{\partial x} f(x, u) \Big|_{x=x^{(i)}} + \alpha b(x) \geq 0, \\ & \quad \forall x^{(i)} \in \bar{X}, \\ & \quad -b(x) + \sigma_1 c(x) \in \Sigma[x], \\ & \quad \|H_b\| = 1. \end{aligned} \quad (\text{SCSOS-CBF})$$

Proof. Condition $\text{sd}_{\text{ov}}(x^{(i)}) \leq b(x^{(i)})$ indicates that the control invariant set is a subset of safe set in the sense of scenarios $x^{(i)} \in \bar{X}$. The condition $\frac{\partial b(x)}{\partial x} f(x, u) + \alpha b(x) \geq 0$ leads to a convex problem when seeking barrier functions with numerical means. Condition $\forall x^{(i)} \in \bar{X}$ restricts the sampling points are in the reachable set \mathcal{C} . Condition $-b(x) + \sigma_1 c(x) \in \Sigma[x]$ indicates that for any x , $-b(x) + \sigma_1 c(x) \geq 0$ and further $\forall x \in \mathcal{B}, c(x) \geq 0$. Regularization of H_b enables to enlarge the volume of the set \mathcal{B} by maximizing d_b . \square

Here we note that our method does not require to parameterize the controller, unlike the results in literature [16]. The objective is to maximize the value of d_b to enlarge the volume of the control invariant set \mathcal{B} , under the regularization of H_b . The decision variables u, σ_1, H_b, d_b are stacked by $z \in \mathbb{R}^e$ for the ease of following theoretic analysis, where $e = m + 1 + n^2 + 1$. The set of satisfying all the constraints about scenarios is defined by \mathcal{S}_x . Note, the constraint $b \in \mathcal{B} : \|H_b\| = 1$ does not belong to \mathcal{S}_x , since it is independent of scenarios \bar{X} .

Algorithm 1 is a data-driven methodology for CBF synthesis considering the system dynamics. Once the SOS program is built, we can solve (SCSOS-CBF) with a semi-definite programming solver. When the next state x^{k+1} comes in, we only need to go through lines 2-7 in Algorithm 1. Although the proposed algorithm costs more time to construct the CBF due to the multi-sampling process, it aims to enlarge the control invariant set \mathcal{B} for the specific mesh shapes of the manipulators and obstacles.

D. Probabilistic Guarantee for Safety

Since the number of sampling is finite, the data-driven method cannot guarantee the equivalence between $b(x^{(i)}) \leq \text{sd}_{\text{ov}}(x^{(i)})$ and $b(x) \leq \text{sd}_{\text{ov}}(x)$. 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 , due to the uncertainty caused by sampling. Violation is defined as $\{\delta \in \Delta : z \notin \mathcal{S}_x\}$. A necessary notion termed *violation probability* is given as:

Definition 3 (violation probability [15]). *The violation probability of a given solution z is defined as $V(z) = \mathbb{P}\{x \in \mathcal{C} : z \notin \mathcal{S}_x\}$.*

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

Algorithm 2: Determine the number of samples

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

```

1  $\bar{N}_{\min} = e$ .
2 while  $\bar{N}_{\min} + 1 \leq \bar{N}_{\max}$  do
3   compute the inverse incomplete beta function
      →  $\gamma_L$  and →  $\gamma_U$ , based on  $(\beta, e, \bar{N}_{\min} - e + 1)$ 
      and  $(\beta, e, \bar{N}_{\max} - e + 1)$ , respectively.
4    $t_{L1} = 1 - \gamma_L$ ,  $t_{L2} = 1$ ,  $t_{U1} = 1 - \gamma_U$ ,  $t_{U2} = 1$ .
5   obtain  $P_{L1}$   $P_{L2}$   $P_{U1}$  and  $P_{U2}$  according to (12)
      based on  $t_{L1}$ ,  $t_{L2}$ ,  $t_{U1}$  and  $t_{U2}$ .
6   if  $P_{L1} \times P_{L2} \geq 0$  then
7     |  $\epsilon_1 = 0$ .
8   else
9     | while  $t_{L2} - t_{L1} > 0$  do
10    |   |  $t_L = \lceil(t_{L2} + t_{L1})/2\rceil$ .
11    |   | obtain  $P_{tL}$  according to (12) 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    |   |  $\epsilon_1 = 1 - t_{L2}$ .
18  | end
19 end
20 compute  $\epsilon_2$  based on  $t_{U1}$ ,  $t_{U2}$ ,  $P_{U1}$  and  $P_{U2}$ .
21 if  $|\epsilon_1 - \epsilon| > \epsilon_p$  or  $|\epsilon_1 - \epsilon_2| > \epsilon_p$  then
22   |  $\bar{N}_{\min} = \bar{N}_{\min} + \lceil(\bar{N}_{\min} - \bar{N}_{\max})/2\rceil$ 
23 else
24   |  $\bar{N} = \bar{N}_{\min}$ .
25   | return  $\bar{N}$ 
26 end
27 end

```

Definition 4. 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.

Then, the probabilistic result of violation probability based on the \bar{N} scenarios is given in the following results.

Theorem 3 (adopted from [25]). *Consider the CBF synthesis program (SCSOS-CBF) with $\bar{N} > e = m + n^2 + 2$. Given*

confidence parameter $\beta \in (0, 1)$, for any $k = 0, 1, \dots, e$, consider the polynomial equation in the t variable

$$\binom{\bar{N}}{k} t^{\bar{N}-k} - \frac{\beta}{2\bar{N}} \sum_{i=k}^{\bar{N}-1} \binom{i}{k} t^{i-k} = \frac{\beta}{6\bar{N}} \sum_{i=\bar{N}+1}^{4\bar{N}} \binom{i}{k} t^{i-k}. \quad (12)$$

This equation has two solutions in $[0, +\infty)$ which are denoted by $\underline{t}(k)$ and $\bar{t}(k)$, where $\underline{t}(k) \leq \bar{t}(k)$. Let $\underline{\epsilon}(k) := \max\{0, 1 - \bar{t}(k)\}$ and $\bar{\epsilon} := 1 - \underline{t}(k)$. Then, for \mathcal{C} and any π , it holds that

$$\mathbb{P}^{\bar{N}}\{\underline{\epsilon}(c_{\bar{N}}^*) \leq V(z^*) \leq \bar{\epsilon}(c_{\bar{N}}^*)\} \geq 1 - \beta. \quad (13)$$

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 3.

Corollary 3.1. *For 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 $\mathbb{P}^{\bar{N}}\{V(z^*) \leq \bar{\epsilon}(c_{\bar{N}}^*)\} \geq 1 - \beta$.*

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

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

$$\bar{N}(\epsilon, \beta) \geq \begin{cases} \arg \min_{\bar{N} \in \mathbb{N}} \bar{N} \\ \text{s.t. (12)} \end{cases} \quad (14)$$

Although Lemma 3 gives guidance on how many sampled data are necessary for the acceptable risk level and confidence, the result is hard to obtain since the optimization problem in (14) is a non-convex 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 risks. Line 21 gives the terminal conditions: 1) the risk based on \bar{N} samples is near the risk goal, 2) the risk cannot decrease too much with \bar{N} increasing.

IV. SIMULATION AND IMPLEMENTATION

This section gives the control synthesis with the proposed CBF construction to formulate a safety-critical control for the robotic manipulator.

A. Control Synthesis

We pre-construct a trajectory as a series of way points by planning methods for the manipulators, but it is unnecessarily safe. u_{des} in (SCB-QP) is computed with a P controller to

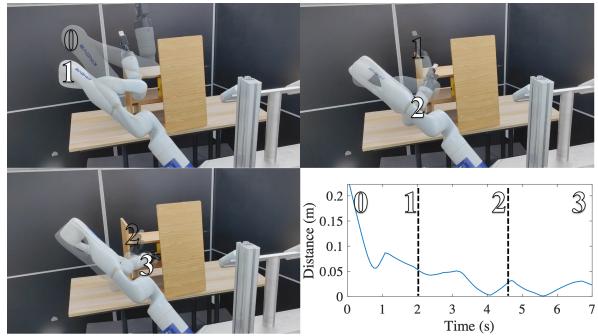


Fig. 2: Experiment implemented on the Kinova Gen3, reaching an object on the left side

the next way-point. Similar to [7], we also set a way-point switch mechanism to avoid the manipulator getting stuck. As pointed out in [7, Proposition 1], (SCB-QP) can guarantee safety with kinematic model of robotic manipulators when using an exponential stable low-level velocity tracking controller. Specifically, the states x are the configuration of each joint, and the inputs u_k are the velocity of each joint.

Definition 5 (Low-level velocity tracking controller). *The manipulator is embedded a low-level velocity tracking controller. For a velocity command $v_c(q; t)$, consider the corresponding tracking error:*

$$\dot{e} = \dot{q} - v_c(q; t); \quad (15)$$

The controller $u = k(x; t)$ can exponentially stabilize the tracking:

$$\|\dot{e}(t)\|_2 \leq l e^{-\lambda t} \|\dot{e}_0\| \quad (16)$$

where $l, \lambda > 0$.

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

B. Implementation

We implement our methodology for manipulator motion planning 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 experimental scenarios have real scenes (see Fig.2 and Fig.3). We aim at using a Kinova Gen 3 robotic manipulator to grasp an object behind a board on a shelf. Our method first requires determining the number of

TABLE II: Theoretical risk and computation times of the data-driven CBF-construction

Theoretical risk	Number of samplings	Computation time (ms)
0.5	108	27.41
0.4	138	27.98
0.3	188	25.25
0.2	288	24.49
0.1	588	27.55
0.05	1188	28.28

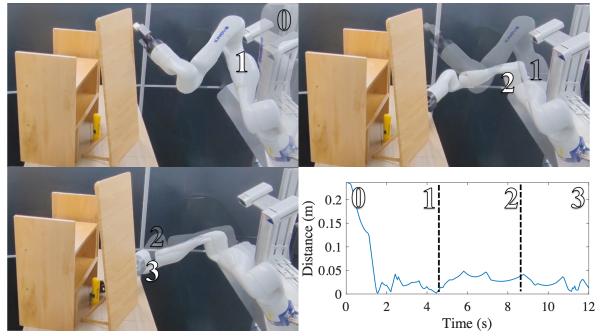


Fig. 3: Experiment implemented on the Kinova Gen3, reaching an object on the right side

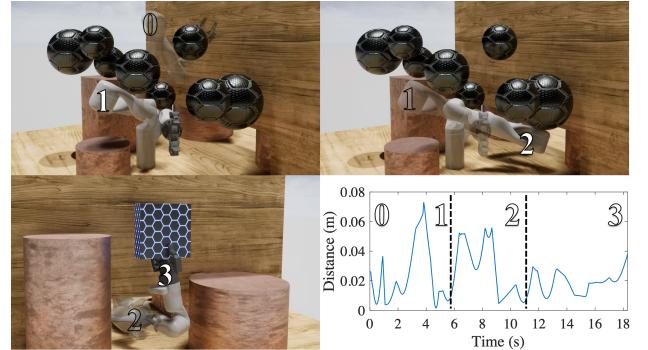


Fig. 4: Experiment implemented on the Kinova Gen3 in a virtual environment with static obstacles

samples needed to construct (SCSOS-CBF) by Algorithm 2. Then, (SCSOS-CBF) is solved with *Sedumi* and *SOSTOOL* [26] 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(R) Core(TM) i9-9980XE CPU, 3.00GHz processor and 64GB RAM.

In the real scene, the primary obstacle of concern is the shelf made of eight boards which blocks 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 tense indoor environment. For example, use a manipulator to grab and unplug a charger behind a LCD monitor. When performing such tasks, the clearance between the manipulator and the obstacles is less than a few millimetres. Our methods can achieve such tasks since the SDF is not linearized, thus enlarge the volume of the feasible space. The data-driven CBF further guarantees safety for the certain model (1). Fig.2 and Fig.3 show the motions and the distances throughout the experiment.

We then design a much more complex environment in the virtual scene to test the 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 hole at the bottom of the board while avoiding collision. Our method can achieve this task with a clearance of one millimetre. Fig.4 show the virtual motion and the distance throughout the trajectory. We continued to simulate

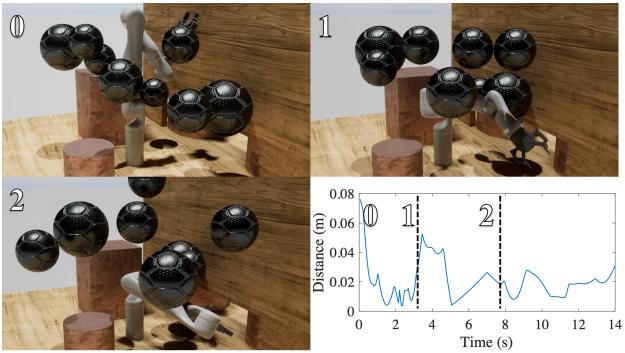


Fig. 5: Experiment implemented on the Kinova Gen3 in a virtual environment with moving obstacles

the virtual scene where the eight balls are moving. Fig.5 shows the virtual test with moving obstacles and the distance throughout the motion. The video can be found at https://www.youtube.com/watch?v=YEYDnT_Ayq6c.

Each scene is tested for ten times with different number of samples. Consequently, the theoretical safety probability is quite different, but *all the tests are successfully operated with safety in practice*. In algorithm 2, we set the $\beta = 0.05$, ϵ from 0.5 to 0.05. The computation time of our data-driven CBF-construction method is shown in tab.II. The theoretical risk decreases as the number of samplings grows. Using 1188 samples, it has a 95% probability that the task is safe. The computation time does not increase with the number of sampling, which shows that our method can guarantee safety and computing timely. As for the average QP computation time, it is 12.41 ms. For comparison purposes, we also conducted the test based on CBF [7] and TrajOpt. The average computation time for QP in CBF is 12.34 ms, which is similar to our result. The average computation time for TrajOpt is 270 ms. The CBF and TrajOpt cannot be successful in all tests, especially when the clearance is with in millimeters. The average computation time for our method is 39.31 ms.

V. CONCLUSION

In this letter, a data-driven CBF construction method is proposed. The method tails the characteristic of the system model to ensure safety under the merit of control invariance. By sampling data in real-time and incorporating the samplings as scenarios in the optimization problem, our method provides less conservative results with a probability guarantee. We also validate the proposed algorithm on industrial robotics manipulators to perform several safety-critical control tasks. In the future, we will explore the multi-manipulator cooperatively equipment assemble tasks.

REFERENCES

- [1] 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 T-ASE*, vol. 19, no. 1, pp. 191–206, 2022.
- [2] 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.
- [3] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [4] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, “Motion planning with sequential convex optimization and convex collision checking,” *IJRR*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [5] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, “Chomp: Gradient optimization techniques for efficient motion planning,” in *2009 IEEE ICRA*, pp. 489–494, IEEE, 2009.
- [6] I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE RAM*, vol. 19, no. 4, pp. 72–82, 2012.
- [7] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, “Safety-critical manipulation for collision-free food preparation,” *IEEE RA-L*, vol. 7, no. 4, pp. 10954–10961, 2022.
- [8] H. Ding, G. Reißig, D. Groß, and O. Stursberg, “Mixed-integer programming for optimal path planning of robotic manipulators,” in *2011 IEEE CASE*, pp. 133–138, IEEE, 2011.
- [9] Y. Yin, S. Hosoe, T. Sugimoto, F. Asano, and Z. Luo, “Hybrid system modeling and control of multi-contact hand manipulation,” in *2004 IEEE ROBIO*, pp. 107–112, IEEE, 2004.
- [10] X. Zhang, A. Liniger, and F. Borrelli, “Optimization-based collision avoidance,” *IEEE TCST*, vol. 29, no. 3, pp. 972–983, 2020.
- [11] J. Pankert and M. Hutter, “Perceptive model predictive control for continuous mobile manipulation,” *IEEE RA-L*, vol. 5, no. 4, pp. 6177–6184, 2020.
- [12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE TAC*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [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 Journal on Robotics and Automation*, vol. 4, no. 2, pp. 193–203, 1988.
- [14] G. Van Den Bergen, “Proximity queries and penetration depth computation on 3D game objects,” in *Game developers conference*, vol. 170, 2001.
- [15] M. C. Campi and S. Garatti, “The exact feasibility of randomized solutions of uncertain convex programs,” *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.
- [16] H. Wang, K. Margellos, and A. Papachristodoulou, “Safe controlled invariance for linear systems using sum-of-squares programming,” *arXiv preprint arXiv:2207.00321*, 2022.
- [17] M. Korda, D. Henrion, and C. N. Jones, “Convex computation of the maximum controlled invariant set for polynomial control systems,” *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2944–2969, 2014.
- [18] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, “Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning,” in *2017 IEEE/RSJ IROS*, pp. 1366–1373, IEEE, 2017.
- [19] L. Han, F. Gao, B. Zhou, and S. Shen, “Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots,” in *2019 IEEE/RSJ IROS*, pp. 4423–4430, IEEE, 2019.
- [20] A. Thirugnanam, J. Zeng, and K. Sreenath, “Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions,” in *IEEE ICRA*, 2022.
- [21] H. Wang, K. Margellos, and A. Papachristodoulou, “Safety verification and controller synthesis for systems with input constraints,” *arXiv preprint arXiv:2204.09386*, 2022.
- [22] 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,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 22, no. 1-3, pp. 650–659, 2015.
- [23] J. Anderson and A. Papachristodoulou, “Advances in computational Lyapunov analysis using sum-of-squares programming,” *Discrete & Continuous Dynamical Systems-B*, vol. 20, no. 8, p. 2361, 2015.
- [24] B. L. Gorissen, İ. Yanıkoglu, and D. den Hertog, “A practical guide to robust optimization,” *Omega*, vol. 53, pp. 124–137, 2015.
- [25] S. Garatti and M. C. Campi, “Risk and complexity in scenario optimization,” *Mathematical Programming*, pp. 1–37, 2019.
- [26] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. Parrilo, “Sostools version 3.00 sum of squares optimization toolbox for matlab,” *arXiv preprint arXiv:1310.4716*, 2013.