

# Configuration-Aware Safe Control for Mobile Robotic Arm with Control Barrier Functions

Fan Ding<sup>1</sup>, Han Wang<sup>2</sup>, Jianping He<sup>1</sup>, Yi Ren<sup>3</sup>, and Yu Zheng<sup>3</sup>

**Abstract**—Collision avoidance is a widely investigated topic in robotic applications. When applying collision avoidance techniques to a mobile robot, how to deal with the spatial structure of the robot still remains a challenge. In this paper, we design a configuration-aware safe control law by solving a Quadratic Programming (QP) with designed Control Barrier Functions (CBFs) constraints, which can safely navigate a mobile robotic arm to a desired region while avoiding collision with environmental obstacles. The advantage of our approach is that it correctly and in an elegant way incorporates the spatial structure of the mobile robotic arm. This is achieved by merging geometric restrictions among mobile robotic arm links into CBFs constraints. Simulations on a rigid rod and the modeled mobile robotic arm are performed to verify the feasibility and time-efficiency of proposed method. Numerical results about the time consuming for different degrees of freedom illustrate that our method scales well with dimension.

## I. INTRODUCTION

Mobile robotic arm, consisting of robotic arms and mobile or drone bases have gained popularity in robotics community for their wide applications in navigating and grasping tasks with obstacle avoidance requirements. Mobile robotic arms allow more flexible operations to complete assigned tasks and, on the other hand, introduce higher computational complexity because of their spatial structures. Various techniques have been proposed in the past decades for mobile robots collision avoidance problems, including Cell Decomposition [1], Potential Field Methods [2]) and heuristic methods (Fuzzy Logic Controller Technique [3], Neural Network Technique [4]). The readers are referred to [5] for an overview in this area. When applying these collision avoidance techniques to a mobile robotic arm, how to deal with its spatial structure still remains a challenge.

Existing collision avoidance algorithms mainly use geometric envelopes to simplify the spatial structure of the robot and reduce the complexity of the optimization. [6] enveloped the mobile robot and moving obstacles by spheres with different radii in and a humanoid torso robot which is abstracted as a cylinder to navigate among modeled pedestrians was proposed in [7]. Ignoring the spacial structured mobile robots with geometric envelopes restricts the volume of permitted workspace and limits the close interaction between mobile robots and obstacles in a clustered environment.

When considering the spatial structure of the mobile robotic arm, the rigid body links are of great interest in

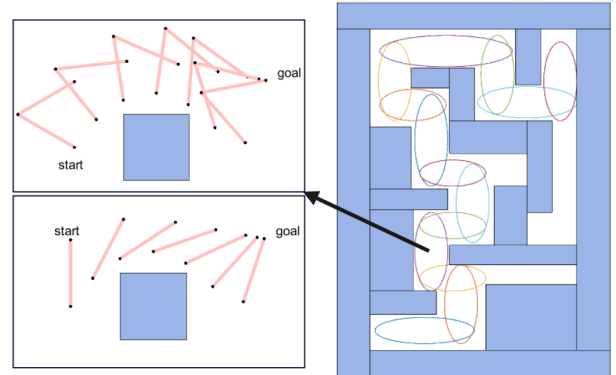


Fig. 1. Scenario of task scene and expected realization. The maze environment in the right figure serves as the configuration space, in which a series of convex ellipsoids are generated as a connected corridor from the start place to the goal region. In the corner, for instance, where the black arrow starts, we expect the rigid rod and the double link bypass the obstacle without collision. Note that both the rigid rod and the double link possess rigid body restrictions, which are considered and solved in an elegant way in this paper.

literature, where [8] approximated the links of the robotic arm by a set of convex polyhedra while [9] treated them as convex ellipsoids. Caused by the high degrees of freedom, calculating the motion control law for the entire mobile robotic arm can be computationally expensive. In [10], a hybrid teleoperation system was proposed where the operator carries out general control for the mobile base while the collision avoidance task is shifted to sensor-based robotic arm subsystem. This decouples the motion constraints between the mobile base and the robotic arm. Therefore, we propose to find a way of incorporating the spatial structure of the mobile robotic arm while conducting collision avoidance task, treating the mobile robotic arm as a whole.

Recently, the Control Barrier Functions (CBFs) approach is proposed as a promising and efficient tool for safe control [11]. By using a quadratic programming based framework the safety requirements can be encoded as affine constraints. The CBFs method is flexible for an arbitrary nominal control law, which determines the desired behavior. Significant success for this approach has been shown in robotics applications, especially in collision avoidance [12]. Extensions such as considering actuation capacity [13], safety-critical Lagrangian systems [14], MPC-CBF [15] for collision avoidance have been proposed. Although these methods are shown to be mathematically rigorous, there is a theory-reality gap dual to the spatial structure of both robots and obstacles.

In this paper, we aim at synthesizing a configuration-aware safe control law for a mobile robotic arm with considered

<sup>1</sup>The Department of Automation, Shanghai Jiao Tong University, Shanghai, China. E-mails: {ding0106, jphe}@sjtu.edu.cn

<sup>2</sup>The Department of Engineering Science, University of Oxford, Oxford, United Kingdom. E-mail: {han.wang}@eng.ox.ac.uk

<sup>3</sup>The Tencent Robotics X Lab, Shenzhen, China. E-mails: {evanyren, petezheng}@tencent.com

complex spatial structure. The proposed safe control law is able to drive the mobile robotic arm to the goal region while avoiding collision in a obstacle-clustered environment. The challenges brought by the spatial structure are solved by ensuring limited edge points of the robot to be within a series of connected convex obstacle-free sets. By solving an online CBFs-based QP with limited number of constraints, the synthesis approach is quite efficient and promising. The main contributions are stated as follows:

- We synthesize a configuration-aware control law that can successfully drives our designed mobile robotic arm to the desired region while avoiding collision with environmental obstacles.
- The proposed approach incorporates the spatial structure of the mobile robotic arm by merging geometric restrictions into CBFs constraints, thus allowing to solve the collision avoidance problem by convex program.
- Our approach is proved to scale well with dimension as the degrees of freedom of the mobile robotic arm increases.

The remainder of this paper is organized as follows. Section II introduces the physical structure of the mobile robotic arm. Preliminaries and reach-avoid-grasp task are specified in Section III. Main results of this paper are shown in Section IV. Extensive numerical simulations are presented in Section V. We conclude the paper in Section VI.

## II. STRUCTURE OF THE MOBILE ROBOTIC ARM

Our designed mobile robotic arm is shown in Fig. 2. It consists of a holonomic four-Mecanum-wheel mobile base and an on-board robotic arm, also equipped with a stereo camera on the top of it. NVIDIA TX2 is embedded as the central computing unit. We tailor a 3D-printed shell as the robot body, on which the robotic arm is set to execute the grasping configuration. The equipped robotic arm is Wkata Robotic Arm, which possesses the standard structure of industrial manipulators with six (plus one) revolute joints. The stereo camera (D435) is mounted on the link between joint 3 and joint 4 of the robotic arm, thus constituting the eye-in-hand configuration which provides higher flexibility for robot operations and precise sights of the target [16].

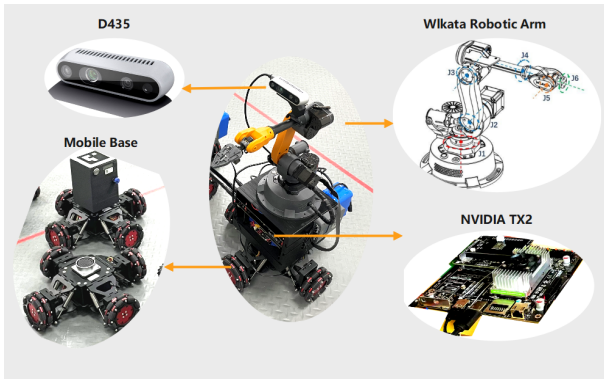


Fig. 2. The systematic structure of the mobile robotic arm.

This stereo camera is an RGB-D camera, from which the RGB information is updated at 30 Hz and the depth information updated at 90 Hz. With the Intel RealSense SDK 2.0 (a provided Intel software), the stereo camera can get on-chip self-calibration. After that, an HSV (Hue, Saturation and Value)-based color segmentation algorithm [17] and morphology open operation are utilized to detect the colored environmental obstacles. Then we manage to get reliable depth information  $Z_c$  of the object in the camera coordinate system with a depth information filter [18]. Let pixel coordinate of point  $m$  on the pixel plane be  $(X_{\text{pixel}}, Y_{\text{pixel}})$ , its real-world coordinate in camera coordinate system  $(X_c, Y_c)$  can be calculated according to the following equations

$$\frac{x_{\text{pixel}} - \text{ppx}}{f_x} = \frac{X_c}{Z_c}, \quad \frac{y_{\text{pixel}} - \text{ppy}}{f_y} = \frac{Y_c}{Z_c},$$

where  $\text{ppx}, \text{ppy}, f_x, f_y$  are the camera intrinsic parameters. Local 3D world reconstruction [19] frame is performed in Fig. 3. Thanks to the high-frequency detection rate of the

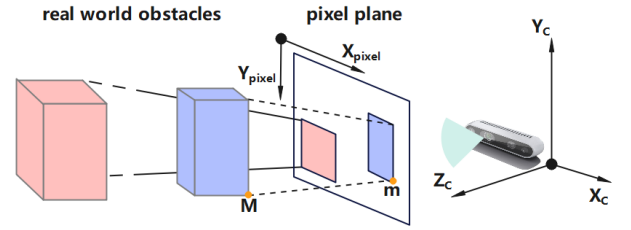


Fig. 3. Real world reconstruction frame. The pink and purple bounding boxes in the pixel plane are the detected results achieved by the HSV-based color segmentation algorithm [17] and minimum volume bounding box method [20]. The corresponding cubes are the reconstructed real-world obstacles

camera, our mobile robotic arm is capable of detecting the surroundings and re-localizing itself in real time. The control flow is illustrated in Fig. 4.

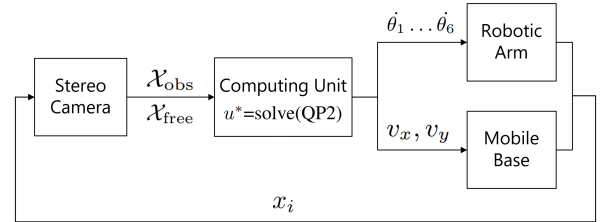


Fig. 4. The control framework of our designed mobile robotic arm. The obstacle-free space  $\mathcal{X}_{\text{free}}$  and the obstacle space  $\mathcal{X}_{\text{obs}}$  are approximated with local 3D real world reconstructed by the stereo camera. Then the computing unit utilizes the obstacle-free space  $\mathcal{X}_{\text{free}}$  and obstacle space  $\mathcal{X}_{\text{obs}}$  information to solve the control law  $u$  from a QP with designed CBF constraints. The control law  $u$  comprises of angle velocities  $\theta_1 \dots \theta_6$  of the robotic arm joints and the velocities  $v_x, v_y$  of the mobile base. The mobile robotic arm state  $x_i$  is updated by movement of the mobile base and rotation of the robotic arm and fed back to the stereo camera.

### III. PRELIMINARIES AND PROBLEM FORMULATION

#### A. Control Barrier Function

The set of real numbers is denoted by  $\mathbb{R}$ , while  $\mathbb{R}^n$  lifts the dimension to  $n$ . Let  $x \in \mathbb{R}^n$  and  $u \in \mathbb{R}^m$  be the state vector and control input, respectively. Consider a nonlinear control affine system as

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

where we suppose that functions  $f(\cdot)$  and  $g(\cdot)$  are both locally Lipschitz continuous. Consider a set  $\mathcal{C}$  defined as the zero-super level set of a continuously differentiable function  $H : \mathbb{R}^n \rightarrow \mathbb{R}$  as

$$\mathcal{C} = \{x \in \mathbb{R}^n : H(x) \geq 0\}, \quad (2)$$

and  $\frac{\partial H(x)}{\partial x} \neq 0$  for all  $x \in \{x | H(x) = 0\}$ .

**Definition 1.** A set  $\mathcal{C}$  is forward invariant with respect to system (1) if for every initial state  $x(t_0) \in \mathcal{C}$ , there exists  $u$ , such that the solutions remain within  $\mathcal{C}$ , i.e.,  $x(t) \in \mathcal{C}$  for  $\forall t \geq t_0$ .

The system (1) is safe with respect to a set  $\mathcal{C}$  if  $\mathcal{C}$  is forward invariant and  $\mathcal{C} \subseteq \mathcal{S}$  where  $\mathcal{S}$  is a compact set defined the safe region. Assuming that  $H(x) = 0$  at  $t = t_0$ , a naturally associated condition to keep the set  $\mathcal{C}$  forward invariant is  $\dot{H}(x) \geq 0$  for any  $t \geq t_0$ . By applying the chain rule,  $\dot{H}(x)$  is specified as:

$$\dot{H}(x) = \frac{dH(x)}{dt} = \frac{\partial H(x)}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial H(x)}{\partial x} \dot{x}. \quad (3)$$

Incorporating (1) into (3), we have

$$\frac{\partial H(x)}{\partial x} (f(x) + g(x)u) \geq 0, \quad (4)$$

which is regarded as the safe constraint. In the real-world applications, this constraint is rather strict that feasible solutions can rarely be obtained and an alternative relaxed formulation was proposed in [11].

**Definition 2.** A continuous function  $\alpha(\cdot) : (-b, a) \rightarrow (-\infty, \infty)$  is said to belong to extended class- $\mathcal{K}$  for some  $a, b > 0$  if it is monotonically increasing and  $\alpha(0) = 0$ .

**Definition 3.** [21] Let  $\mathcal{C} \subseteq D \subset \mathbb{R}^n$  be the set defined by  $H(\cdot)$ . Then  $H$  is a control barrier function if there exists an extended class- $\mathcal{K}$  function  $\alpha$ , such that  $\forall x \in D$ ,  $\exists u \in \mathbb{R}^m$ , the following inequality holds:

$$\sup_{u \in \mathbb{R}^m} \left[ \frac{\partial H(x)}{\partial x} (f(x) + g(x)u) \right] \geq -\alpha(H(x)).$$

Definition 3 renders the set  $K_{CBF}$  of control laws  $u$  that guarantees the safe set  $\mathcal{C}$  forward invariant [22]. The set of safe control law is defined by:

$$K_{CBF} = \left\{ u \in \mathbb{R}^m : \frac{\partial H(x)}{\partial x} (f(x) + g(x)u) \geq -\alpha(H(x)) \right\}. \quad (5)$$

#### B. Homogeneous Transformation Matrix

Let  $P_M^C \in \mathbb{R}^3$  be the position vector of point  $M$  in coordinate  $C$  and  $P_M^B \in \mathbb{R}^3$  be the position vector of point  $M$  in coordinate  $B$ . If the origins of coordinate  $B$  and  $C$  coincide, we can find  $R_B^C \in SO(3) \subset \mathbb{R}^{3 \times 3}$ , such that

$$P_M^C = R_B^C \times P_M^B,$$

where each column of  $R_B^C$  represents the projections of the three axes of coordinate  $B$  on each axis of the coordinate  $C$ . If the origins of coordinate  $B$  and  $C$  do not coincide, then we can derive a homogeneous transformation matrix  $T_B^C \in SE(3) \subset \mathbb{R}^{4 \times 4}$ :

$$T_B^C = \begin{bmatrix} R_B^C & p_B^C \\ 0 & 1 \end{bmatrix}, \quad (6)$$

where  $p_B^C \in \mathbb{R}^3$  denotes the translation offset from the origin of coordinate  $B$  to the origin of coordinate  $C$ , such that

$$\begin{bmatrix} P_M^C \\ 1 \end{bmatrix} = T_B^C \times \begin{bmatrix} P_M^B \\ 1 \end{bmatrix}.$$

#### C. Reach-Avoid-Grasp Problem Formulation

The goal of this paper is to find a feasible control law that drives our designed mobile robotic arm to the goal region while avoiding collision with obstacles. A region is regarded as the goal region if, in which the mobile robotic arm can execute the grasp configuration, i.e. the mounted robotic arm could grasp the target requiring no more movement of the mobile base. More specifically, the reach-avoid-grasp problem is posed as follows:

**Problem 1** (Specify Goal Region). Given a mobile robot workspace  $\mathcal{X} \in \mathbb{R}^n$  and a goal position  $x_{\text{goal}} \in \mathcal{X}$  as a priori, find the goal region  $\mathcal{X}_{\text{goal}} \subset \mathcal{X}$  in which our designed mobile robotic arm can execute the grasp configuration.

**Problem 2** (Geometrically Describe the Mobile Robotic Arm). Find  $\mathcal{M}$  proper edge points  $a_i$ ,  $i = 1 \dots \mathcal{M}$ , and assign  $\mathcal{M} - 1$  line segments  $A_i$ ,  $i = 1 \dots \mathcal{M} - 1$ , where  $A_i$  connects points  $a_i$  and  $a_{i+1}$  to represent the mobile robotic arm  $\mathcal{A}$ :

$$\mathcal{A} = \bigcup_{i=1}^{\mathcal{M}-1} A_i.$$

**Problem 3** (Find the Safe Control Law). Given a mobile robot workspace  $\mathcal{X} \in \mathbb{R}^n$ , and divide it into obstacle-occupied space  $\mathcal{X}_{\text{obs}} \in \mathcal{X}$  and obstacle-free space  $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$ , find a feasible control law that drives the mobile robotic arm  $\mathcal{A}$  from initial state  $x_0 \in \mathcal{X}_{\text{free}}$  to a state in the goal region  $\mathcal{X}_{\text{goal}} \subset \mathcal{X}_{\text{free}}$  while avoiding collision with the obstacles, i.e.

$$\mathcal{A}(t) \subset \mathcal{X}_{\text{free}}.$$

## IV. MAIN RESULTS

### A. Goal Region and Grasping

The key to find the goal region  $\mathcal{X}_{goal}$  (i.e. solve Problem 1) for the mobile robotic arm is to specify the workspace  $W$  of the robotic arm. The workspace of a robotic arm represents the region where its end-effector can reach without violating the joint limits. In this paper, we utilize kinematic modeling to numerically describe the workspace of the robotic arm. Physical model with axis assignment is shown in Fig. 5(a) and the DH (Denavit-Hartenberg) parameters [23] are obtained according to the axis assignment, shown in Table I. The workspace of the robotic arm, which is

TABLE I  
MODIFIED DH MODEL PARAMETERS

coordinate $i$	$\theta_i$	$b_i$	$a_{i-1}$	$\alpha_{i-1}$	Rotation Range
axis1	0	80	0	0	-100° to +100°
axis2	$-\pi/2$	0	32	$-\pi/2$	-60° to +90°
axis3	0	0	108	0	-180° to +50°
axis4	0	176	20	$-\pi/2$	-180° to +180°
axis5	$\pi/2$	0	0	$\pi/2$	-180° to +40°
axis6	0	-20	0	$\pi/2$	-180° to +180°

visualized by yellow point cloud and shown in Fig. 5(b), is obtained by applying the forward kinematics provided by Robotics Toolbox in Matlab. Since the workspace is relevant to the state  $x$  of the mobile robotic arm, we rewrite the workspace as  $W(x)$ . Given the goal position  $x_{goal}$ , the goal region is defined as

$$\mathcal{X}_{goal} = \{x \in \mathcal{X}_{free} : x_{goal} \in W(x)\}. \quad (7)$$

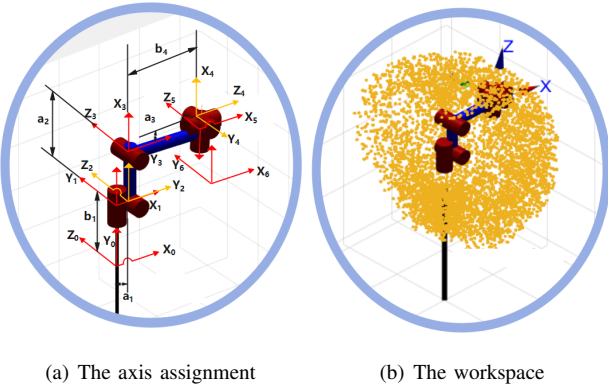


Fig. 5. The axis assignment and workspace of the physical model of our robotic arm. In Fig. 5(a), six revolute joints are denoted where blue and red cylinders meet. The blue and red cylinders represent the rigid link between the joints. DH parameters  $a_1, a_2, a_3$  and  $b_{1,4}$  are labeled with double arrows. The yellow point cloud in Fig. 5(b) numerically describes the workspace of the robotic arm. Note that the workspace maintains its spatial configuration while translating and rotating with the movement of the robotic arm base.

As for grasping configuration, we consider the homogeneous transformation matrix [24],  $T_c^a$  from the camera coordinate  $(X_c Y_c Z_c)$  to the robotic arm coordinate  $(X_a Y_a Z_a)$ .

According to (6), we have

$$\begin{bmatrix} X_a & Y_a & Z_a & 1 \end{bmatrix}^T = T_c^a \begin{bmatrix} X_c & Y_c & Z_c & 1 \end{bmatrix}^T, \quad (8)$$

where  $T_c^a$  is specified as (9) calculated with the geometric relationship in Fig. 6.

$$T_c^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & -\sqrt{l_1^2 + l_2^2} \cos(\alpha + \theta) \\ 0 & -\sin \alpha & \cos \alpha & \sqrt{l_1^2 + l_2^2} \sin(\alpha + \theta) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Therefore, when the object is detected by our camera, its

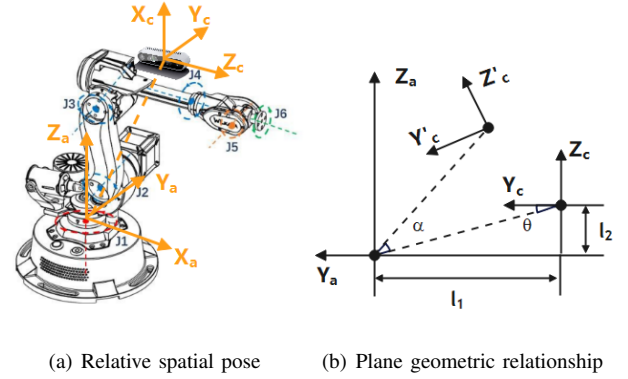


Fig. 6. Geometric relationship between camera coordinate and arm coordinate. 6(a) shows the relative spatial pose between camera coordinate and arm coordinate on the robotic arm. Joint 1, joint 2, joint 3, joint 4, joint 5 and joint 6 are abbreviated as J1, J2, J3, J4, J5 and J6. Geometric relationship between these two coordinates is extracted as 6(b) where axis  $Z_c$  and  $Y_c$  rotate to  $Z'_c$  and  $Y'_c$  by rotation (defined as angle  $\alpha$ ) of J3.

coordinate in robotic arm coordinate can be derived from (8) and (9). Then with inverse kinematics tool of Robotics Toolbox in matlab, we can specify the grasp configuration, i.e. angle assignment of each joint.

### B. Solution of Reach-Avoid Problem

The reach-avoid task comprises desired position reaching and obstacle avoiding steps. By deriving the workspace of the robotic arm, the desired position reaching step is relaxed to desired region reaching step, which is less computationally expensive.

We now take advantage of the control framework (shown in Fig. 4) to carry out the reach-avoid task. When the mobile robotic arm arrives at a position  $x_i$ , the stereo camera (D435) captures present perspective of surroundings. By applying local 3D world reconstruction method [19], obstacle-free space  $\mathcal{X}_{free}$  and obstacle-occupied space  $\mathcal{X}_{obs}$  can be properly approximated. Generally speaking, the obstacle-free space is a non-convex region [25], [26], our approach considers a series of convex sets  $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_N$  (i.e. ellipsoids, convex polygons) extracted from the obstacle-free space. Based on  $\mathcal{X}_{free}$  and  $\mathcal{X}_{obs}$ , proper convex set  $\mathcal{C}_i$  enclosing current state  $x_i$  can be found by solving convex optimization problems [27]. We assign succeeded state  $x_{i+1}$  of  $x_i$  such that the

distance between  $x_i$  and  $x_{i+1}$  is maximized, i.e. solving

$$\begin{aligned} \max \quad & \|x_i - x_{i+1}\|_2 \\ \text{s.t.} \quad & x_{i+1} \in \mathcal{C}_i \subset \mathcal{X}_{\text{free}}, \end{aligned} \quad (\text{QP1})$$

When considering the spatial structure of our designed mobile robotic arm, we mainly focus on the rigid body links (link 1 between J1 and J2, link 2 between J2 and J3, link 3 between J3 and J5, link 4 between J5 and J6 shown in Fig. 6(a)) of the robotic arm. To solve Problem 2, we choose the central points of J1, J2, J3, J5, J6 as the edge points  $a_i$ ,  $i = 1 \dots 5$ . And abstract the rigid body link of the robotic arm as line segments  $A_i$ ,  $i = 1, \dots, 4$ , connecting the edge points  $a_i$  and  $a_{i+1}$ , to represent the robotic arm  $\mathcal{A}$ , where

$$\mathcal{A} = \bigcup_{i=1}^4 A_i.$$

In a more general sense, we denote  $N_l$  as the number of the abstracted line segments and  $N_{ep}$  as the number of the edge points. The mobile robotic arm is therefore generalized as:

$$\mathcal{A} = \bigcup_{i=1}^{N_l} A_i.$$

**Theorem 1.** *To keep the mobile robotic arm safe in the course of reach-avoid task, we only need to keep the chosen corner points safe, i.e., the safety control is guaranteed by the following control law*

$$u^*(x) = \arg \min_{u \in \cap_{k=1, \dots, N_{ep}} K_{CBF}^{(k)}} \|u - u^p(x)\|_2, \quad (\text{QP2})$$

where  $u^p(x) = k_p(x_{i+1} - x_i)$  and  $k_p$  is the tuned proportional gain.

*Proof:* First we prove that the edge points are safe by applying the safe control law. Let  $x^{(k)}$  be the state of the edge point  $k$ , extended from (5), we obtain the  $K_{CBF}^{(k)}$  with respect to  $x^{(k)}$  as:

$$K_{CBF}^{(k)} = \left\{ u \in \mathbb{R}^m : \left[ \frac{\partial H(x)}{\partial x} (f(x) + g(x)u) + \alpha(H(x)) \right] \Big|_{x=x^{(k)}} \geq 0 \right\}. \quad (10)$$

Given the states  $x^{(k)}$ ,  $x^{(k+1)}$  of edge point  $k$ , edge point  $k+1$ , a set  $\mathcal{C}$  by (2) and  $u \in K_{CBF}^{(k)} \cap K_{CBF}^{(k+1)}$ , then  $\mathcal{C}$  is ensured forward invariant with (5) [22]. According to Definition 1, we can derive that  $x^{(k)}$  and  $x^{(k+1)}$  are ensured within set  $\mathcal{C}$  (i.e. edge point  $k$  and edge point  $k+1$  are safe). Then we prove that the mobile robotic arm  $\mathcal{A}$  is safe given all the edge points safe. For a convex set  $\mathcal{C}_i \subset \mathbb{R}^n$ ,  $\forall y^{(1)}, y^{(2)} \in \mathcal{C}_i$ , equation

$$\lambda y^{(1)} + (1 - \lambda) y^{(2)} \in \mathcal{C}_i, \forall \lambda \in [0, 1]$$

holds according to convex set definition. This indicates that if points  $y^{(1)}$  and  $y^{(2)}$  are within the convex set, then the generalized line segment connecting them is ensured in this convex set. Earlier in this subsection, we get a series of convex sets  $\mathcal{C}_0, \dots, \mathcal{C}_N$  from obstacle free space

$\mathcal{X}_{\text{free}} \subset \mathbb{R}^n$  and now we substitute  $\mathcal{C}$  with one of the convex set. Therefore, given  $a_i \in \mathcal{C}$ ,  $i = 1, \dots, N_{ep}$  (edge points are safe), line segments  $A_i \subset \mathcal{C}(\text{safe})$ ,  $i = 1, \dots, N_l$  (line segments are safe), hence  $\mathcal{A} = \bigcup_{i=1}^{N_l} A_i \subset \mathcal{C}$  (the mobile robotic arm is safe).  $\square$

Based on the extracted convex set  $\mathcal{C}_0, \dots, \mathcal{C}_N$ , the corresponding control barrier function  $H_0(x), \dots, H_N(x)$  can be specified by

$$\mathcal{C}_i = \{x \in \mathbb{R}^n : H_i(x) \geq 0\}, i = 0, \dots, N.$$

The corresponding safe control law  $u^*(x)$  is computed by solving (QP2) with substituting  $H(x)$  by the specific  $H_i(x)$ , where  $i$  depends on the current located safe set. For now we have solved Problem 3.

## V. NUMERICAL SIMULATION

In this section, simulation of a rigid rod and a mobile robotic arm in a complex maze is performed to verify the proposed algorithm. The QP solver is CVXGEN [28].

Firstly we apply our method on a rigid rod, aiming at driving the rigid rod from the initial configuration to the goal region without collision with the rectangle obstacles in a maze environment. The maze environment introduces a start position on the bottom left and a goal position on the top right, shown in Fig. 7.

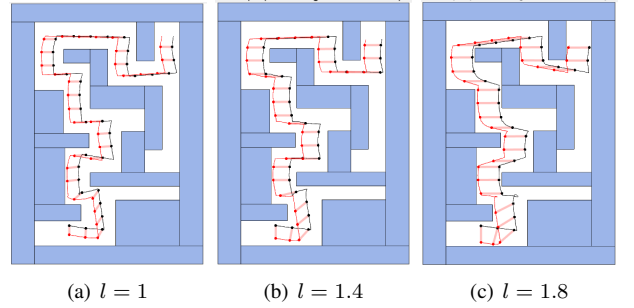


Fig. 7. Simulation results of a rigid rod in a maze environment with length  $l = 1$  in Fig. 7(a),  $l = 1.4$  in Fig. 7(b) and  $l = 1.8$  in Fig. 7(c). The rigid rod (the thick pink segment with a black vertex and a red vertex) aims at navigating from start position to the goal position without collision with obstacles in the maze environment. Slim red curve and black curve are the trajectories of the red vertex and the black vertex. In both cases with different initial pose, the rigid rod successfully and safely reaches the goal region.

Then we conduct simulations on our designed mobile robotic arm. While describing the workspace of the robotic arm, we notice that J4 and J6 play little role in forming the workspace, both of which are for adjusting the pose of the end-effector to better execute the grasp configuration. Therefore, J4 and J6 make little difference in the collision avoidance problem and we focus on J1, J2, J3 and J5 while modeling the mobile robotic arm as shown in Fig. 8. When four joints are considered, the modeled mobile robotic arm is theoretically close to our designed mobile robotic arm in real world. According to Theorem 1, each joint corresponds to one CBF constraint in a convex quadratic programming



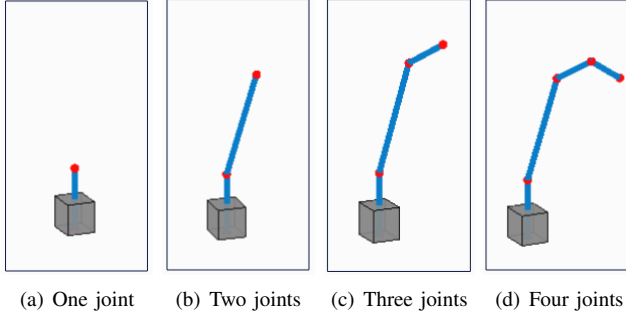


Fig. 8. Models of mobile robotic arm considering different amount of rotating joints. Mobile base is modeled as a grey cube and rigid body links are modeled as blue line segments in Fig. 8(a)-8(d). Red points represent the edge points on the mobile robotic arm, and one edge point is sheltered in the mobile base. One joint (J1), two joints (J1, J2), three joints (J1, J2, J3) and four joints (J1, J2, J3, J5) are respectively considered in Fig. 8(a), 8(b), 8(c) and 8(d).

framework, the algorithm therefore scales well with dimension. The time consuming of our approach is enumerated as  $0.0220(s)$ ,  $0.0245(s)$ ,  $0.0279(s)$  and  $0.0292(s)$  when one, two, three and four joint(s) are respectively considered in our simulation.

Now we present the simulation results of considering three-joint and four-joint mobile robotic arm model. We refine the state vector  $x$  and input vector  $u$  in the state space model of the mobile robotic arm as:

$$x = [X \ Y \ Z]^T, u = [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3 \ \dot{\theta}_5 \ v_x \ v_y]^T,$$

where  $X, Y, Z$  denote the coordinate of the end-effector,  $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_5$  are angular velocities of J1, J2, J3, J5 and  $v_x, v_y$  are velocities of the mobile base. When specifying the state space model of our mobile robotic arm, we only take the most complex situation (four joints) for example, shown as follow

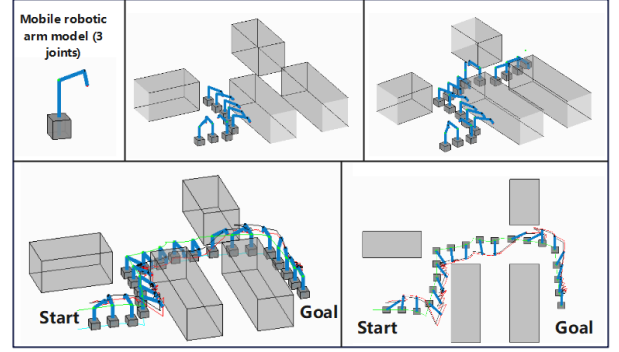
$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & 1 & 0 \\ f_{21} & f_{22} & f_{23} & f_{24} & 0 & 1 \\ 0 & f_{32} & f_{33} & f_{34} & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_5 \\ v_x \\ v_y \end{bmatrix}, \quad (11)$$

where,

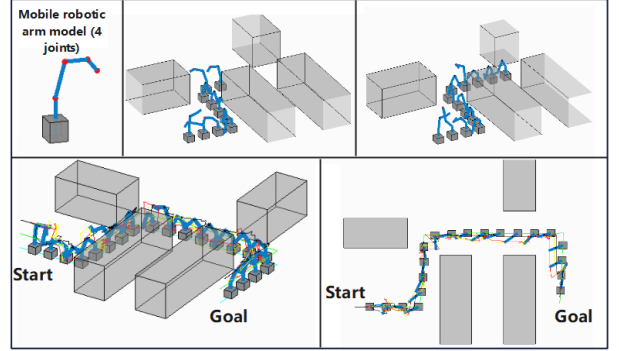
$$\begin{aligned} f_{11} &= -a_2 s_1 s_2 - d_4 s_1 c_{23} + d_6 s_1 c_{235}, \\ f_{12} &= -a_2 c_1 c_2 - d_4 c_1 s_{23} + d_6 c_1 s_{235}, \\ f_{13} &= -d_4 c_1 s_{23} + d_6 c_1 s_{235}, \\ f_{14} &= d_6 c_1 s_{235}, \\ f_{21} &= -a_2 c_1 s_2 + d_4 c_1 s_{23} + d_6 s_1 s_{235}, \\ f_{22} &= a_2 s_1 c_2 - d_4 s_1 s_{23} + d_6 s_1 s_{235}, \\ f_{23} &= -d_4 s_1 s_{23} + d_6 s_1 s_{235}, \\ f_{24} &= d_6 s_1 s_{235}, \\ f_{32} &= -a_2 s_2 - d_4 c_{23} + d_6 c_{235}, \\ f_{33} &= -d_4 c_{23} + d_6 c_{235}, \\ f_{34} &= d_6 c_{235}. \end{aligned}$$

( $s_1$  denotes  $\sin \theta_1$ ,  $c_1$  denotes  $\cos \theta_1$  and  $s_{23}$  denotes  $\sin(\theta_2 + \theta_3)$ .  $a_2, d_1, d_4, d_6$  are DH parameters in Table I.)

The coordinates of the rest edge points can be calculated according to the rigid body restrictions of the mobile robotic arm. Driven by the safe control law solved by (QP2), our mobile robotic arm can successfully avoid collision with obstacles and navigate to the desired region in the environment where several obstacles are randomly set, shown in Fig. 9. Fig. 10 introduces the minimum distance to the safe set boundary as the indicator of safety and the edge points are ensured in the safe set, hence ensuring the whole mobile robotic arm in the safe set as depicted in Theorem 1.



(a) Simulating process of three-joint mobile robotic arm

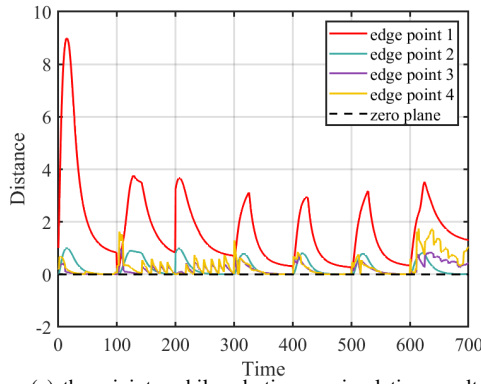


(b) Simulating process of four-joint mobile robotic arm

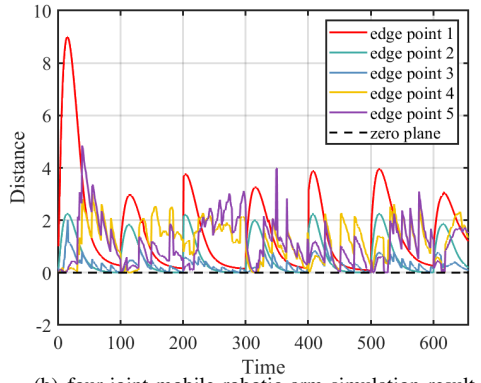
Fig. 9. Simulating process overview of the modeled mobile robotic arm. By applying the safe control law solved by (QP2), the mobile robotic arm manages to reach the goal region while avoiding collision with environmental obstacles both in Fig. 9(a) (three-joint mobile robotic arm) and Fig. 9(b) (four-joint mobile robotic arm).

## VI. CONCLUSION AND FUTURE DIRECTION

In this paper, we solved the problem of configuration-aware safe control for a designed mobile robotic arm with Control Barrier Functions. We firstly propose to geometrically describe our designed mobile robotic arm and mean to fully cooperate the spatial structure of the mobile robotic arm. Then we merge these geometric restrictions into CBFs constraints and solve the safe control law from a QP with designed CBFs constraints. This safe control law can successfully navigate the modeled mobile robotic arm to a desired region while avoiding collision with environmental obstacles.



(a) three-joint mobile robotic arm simulation result



(b) four-joint mobile robotic arm simulation result

Fig. 10. The minimum distance to the safe set boundary of 4 edge points (see Fig. 10(a)) and 5 edge points (see Fig. 10(b)) with respect to time are depicted as curves of different colors. Even though the edge point of the mobile robotic arm may get close to the safe set boundary, the edge point is still ensured safe without collision with the obstacles, also shown in the top view figure in Fig. 9.

We show that our method is promising and computationally efficient by conducting numerical simulations. In the future we will apply our approach to physical test-bed and explore higher dimensional arms.

## REFERENCES

- [1] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1, pp. 467–472, IEEE, 2004.
- [2] L. Tang, S. Dian, G. Gu, K. Zhou, S. Wang, and X. Feng, "A novel potential field method for obstacle avoidance and path planning of mobile robot," in *2010 3rd international conference on computer science and information technology*, vol. 9, pp. 633–637, IEEE, 2010.
- [3] T. S. Hong, D. Nakhaeinia, and B. Karasfi, "Application of fuzzy logic in mobile robot navigation," *Fuzzy Logic-Controls, Concepts, Theories and Applications*, pp. 21–36, 2012.
- [4] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2759–2764, IEEE, 2016.
- [5] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," *Int Rob Auto J*, vol. 2, no. 3, p. 00022, 2017.
- [6] L. Zeng and G. M. Bone, "Mobile robot collision avoidance in human environments," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 41, 2013.
- [7] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.
- [8] C. Chang, M. J. Chung, and B. H. Lee, "Collision avoidance of two general robot manipulators by minimum delay time," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 517–522, 1994.
- [9] M.-Y. Ju, J.-S. Liu, S.-P. Shiang, Y.-R. Chien, K.-S. Hwang, and W.-C. Lee, "Fast and accurate collision detection based on enclosed ellipsoid," *Robotica*, vol. 19, no. 4, pp. 381–394, 2001.
- [10] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in tele-operated whole-sensitive robot arm manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.
- [11] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, pp. 6271–6278, IEEE, 2014.
- [12] P. Grottel, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.
- [13] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.
- [14] F. S. Barbosa, L. Lindemann, D. V. Dimarogonas, and J. Tumova, "Provably safe control of lagrangian systems in obstacle-scattered environments," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 2056–2061, IEEE, 2020.
- [15] A. Thirugnanam, J. Zeng, and K. Sreenath, "Safety-Critical Control and Planning for Obstacle Avoidance between Polytopes with Control Barrier Functions," *arXiv e-prints*, p. arXiv:2109.12313, Sept. 2021.
- [16] G. Dong and Z. Zhu, "Position-based visual servo control of autonomous robotic manipulators," *Acta Astronautica*, vol. 115, pp. 291–302, 2015.
- [17] K. B. Shaik, P. Ganesan, V. Kalist, B. Sathish, and J. M. M. Jenitha, "Comparative study of skin color detection and segmentation in hsv and ycbcr color space," *Procedia Computer Science*, vol. 57, pp. 41–48, 2015.
- [18] M. Camplani and L. Salgado, "Adaptive spatio-temporal filter for low-cost camera depth maps," in *2012 IEEE International Conference on Emerging Signal Processing Applications*, pp. 33–36, IEEE, 2012.
- [19] S. Kagami, Y. Takaoka, Y. Kida, K. Nishiwaki, and T. Kanade, "Online dense local 3d world reconstruction from stereo image sequences," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3858–3863, IEEE, 2005.
- [20] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1628–1633, IEEE, 2008.
- [21] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [22] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, IEEE, 2019.
- [23] E. Farah and S. G. Liu, "Dh parameters and forward kinematics solution for 6d of surgical robot," in *Applied Mechanics and Materials*, vol. 415, pp. 18–22, Trans Tech Publ, 2013.
- [24] R. P. Paul and H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *The International Journal of Robotics Research*, vol. 5, no. 2, pp. 32–44, 1986.
- [25] V. Vasilopoulos and D. E. Koditschek, "Reactive navigation in partially known non-convex environments," in *International Workshop on the Algorithmic Foundations of Robotics*, pp. 406–421, Springer, 2018.
- [26] G. Notomista and M. Saveriano, "Safety of dynamical systems with multiple non-convex unsafe sets using control barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 1136–1141, 2021.
- [27] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI*, pp. 109–124, Springer, 2015.
- [28] J. Mattingley and S. Boyd, "Cvxgen: A code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.