IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

DiffOcclusion: Differentiable Optimization Based Control Barrier Functions for Occlusion-Free Visual Servoing

Shiqing Wei, *Graduate Student Member, IEEE*, Bolun Dai, *Graduate Student Member, IEEE*, Rooholla Khorrambakht, *Student Member, IEEE*, Prashanth Krishnamurthy, *Member, IEEE*, and Farshad Khorrami, *Senior Member, IEEE*

***Abstract*—The visibility (possibly partial) of some image features is crucial to a broad class of visual servoing-based control. In this letter, we consider the setting of image-based visual servoing (IBVS) and address the fundamental problem of keeping a moving object with an unknown motion profile in the field of view while ensuring it remains unobstructed by obstacles. Assuming that the projections of the target and obstacles are both convex polygons, we propose a systematic method for circumscribing these polygons by strictly convex shapes with tunable accuracy. We prove that the minimal scaling factor such that two convex shapes intersect is continuously differentiable with respect to their vertex coordinates. Then, we formulate a control barrier function (CBF) based on this minimal scaling factor and incorporate a motion observer into occlusion-free visual servoing. The effectiveness of our method is validated through both simulation studies and experiments on the Franka Research 3 robotic arm.**

***Index Terms*—Visual servoing, visual tracking, robot safety, sensor-based control.**

I. Introduction

VISION plays an important role in robotic motion planning and control as vision sensors provide geometrical information about the surrounding environment. Specifically, control methodologies that rely on visual measurements for feedback are commonly referred to as visual servoing (VS) [1]. However, the effectiveness of VS-based control methods relies heavily on the quality and the visibility of image features. Therefore, in this work, we address the problem of occlusion-free VS, particularly in scenarios where image features are susceptible to obstruction by other objects.

Manuscript received 22 September 2023; accepted 29 January 2024. Date of publication 9 February 2024; date of current version 23 February 2024. This letter was recommended for publication by Associate Editor P. Di Lillo and Editor J. P. Desai upon evaluation of the reviewers' comments. This work was supported in part by ARO under Grant W911NF-21-1-0155 and Grant W911NF-22-1-0028, and in part by the New York University Abu Dhabi (NYUAD) Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010. *(Corresponding author: Shiqing Wei.)*

The authors are with the Control/Robotics Research Laboratory, Electrical and Computer Engineering Department, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: shiqing.wei@nyu.edu; bolundai@nyu.edu; rk4342@nyu.edu; prashanth.krishnamurthy@nyu.edu; khorrami@nyu.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3364468, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3364468

Our study considers designing occlusion-free controls in the setting of image-based visual servoing (IBVS), also known as VS in the image plane. The primary objective of this work is to address the fundamental challenge of tracking a moving object with an unknown motion profile in the field of view (FoV) while ensuring it remains unobstructed by obstacles. Assuming that the projections of the target and obstacle are both convex polygons, we propose a systematic method for circumscribing these polygons into strictly convex shapes with tunable user-designed *scaling functions*. Further, we prove that the minimal scaling factor such that the two shapes intersect is continuously differentiable with respect to (w.r.t.) their vertex coordinates. We then employ a motion observer to estimate the unknown terms in the motion field equation due to the movement of the target and formulate a control barrier function (CBF) based on this minimal scaling factor to realize occlusion-free VS.

Avoiding occlusion or increasing visibility holds significant importance in numerous vision-based robotic applications [2], [3], [4], [5], [6], [7]. In [2], switching controllers are developed to bypass obstacles and avoid occlusion for non-holonomic robots in a cluttered environment. In [3], the authors compute the minimum distance between a point and a line segment for occlusion avoidance in dual-arm teleoperation. Another approach proposed in [5] avoids collisions and occlusions in quadrotor navigation by representing both quadrotors and obstacles as spheres. These methods commonly represent obstacles using simple geometries such as spheres ([5], [6]), line segments ([3]), or critical points ([4]), which may lack generality or lead to overly conservative results. A distinctive feature of our work is the use of scaling functions, which preserve the obstacle's geometry with tunable accuracy. In addition, our approach can be easily applied to a given set of feature points on the target (instead of the whole target). It is worth mentioning that in case of visual signal loss (e.g., temporary camera breakdown), the method in [8] becomes essential for reconstructing visual features.

Our work is also closely related to differentiable optimization, which deals with optimization problems whose solutions are differentiable w.r.t. the problem configuration parameters. Based on the Karush-Kuhn-Tucker (KKT) conditions and implicit function theorem (IFT), the authors of [9] introduce the differentiable convex optimization layers for disciplined convex programs.

Another related work, [10], presents a differentiable collision detection method for a set of convex primitives, also drawing from IFT and KKT conditions for convex problems. However, the aforementioned methods did not establish the continuous differentiability of the solution. Our previous work, [11], established the continuous differentiability of the minimal scaling factor for strongly convex scaling functions. Our current study demonstrates that continuous differentiability is attainable even under relaxed conditions.

Our contributions: (1) We design and develop one of the first frameworks that address the problem of maintaining a moving object with an unknown motion profile in the FoV while achieving occlusion avoidance using CBFs. (2) We introduce a systematic method for circumscribing convex polygons by strictly convex shapes with adjustable scaling functions, which also generalizes to 3D convex polytopes. (3) Compared with our previous work, we extend the domain of the scaling function from $\mathbb{R}^3$ to $\mathbb{R}^n$ and prove the continuous differentiability of the minimal scaling factor under relaxed conditions (one scaling function being strictly convex and the other convex). (4) We demonstrate the effectiveness of our method through both simulation studies and experiments on the Franka Research 3 (FR3) robotic arm.

Notations: $\|\cdot\|$ is the ℓ_2 norm of a vector or a matrix, and $(\cdot)^\dagger$ is the pseudoinverse of a matrix. I is the identity matrix with proper dimensions. For a $\mathcal{C}^1$ function $h:\mathbb{R}^n\to\mathbb{R}$, its gradient ∇h is a column vector while the partial derivative $\frac{\partial h}{\partial x}$ is a row vector. For a $\mathcal{C}^1$ function $f:\mathbb{R}^n\to\mathbb{R}^m$, the partial derivative $\frac{\partial f}{\partial x}$ is a $m\times n$ matrix.

II. Preliminaries

Consider a control-affine system

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

with $f:\mathbb{R}^{n_x}\to\mathbb{R}^{n_x}$ and $g:\mathbb{R}^{n_x}\to\mathbb{R}^{n_x\times n_u}$ being locally Lipschitz, the states $x\in\mathcal{X}\subset\mathbb{R}^{n_x}$, and the control $u\in\mathcal{U}\subset\mathbb{R}^{n_u}$. Let $h:\mathcal{X}\subset\mathbb{R}^{n_x}\to\mathbb{R}$ be a $\mathcal{C}^1$ function, and define

$$C = \{x\in\mathcal{X} \mid h(x)\geq 0\}, \tag{2a}$$

$$\partial C = \{x\in\mathcal{X} \mid h(x)= 0\}, \tag{2b}$$

$$\mathrm{Int}(C) = \{x\in\mathcal{X} \mid h(x)> 0\} \tag{2c}$$

where C is the *safe set*, ∂C is the boundary of C, and $\mathrm{Int}(C)$ is the interior of C. We assume that $\mathrm{Int}(C)\neq\varnothing$ and $\nabla h(x)\neq 0$ for $x\in\partial C$.

Definition 1 (Control barrier functions [12]): A $\mathcal{C}^1$ function $h:\mathcal{X}\to\mathbb{R}$ is a control barrier function (CBF) if there exists an extended $\mathcal{K}_\infty$ function[1] Γ such that for system (1)

$$\sup_{u\in\mathcal{U}}[L_f h(x) + L_g h(x)u] \geq -\Gamma(h(x)),\quad \forall x\in C \tag{3}$$

where $L_\phi h(\cdot) = \frac{\partial h}{\partial x}(\cdot)\phi(\cdot)$ is the Lie derivative of h w.r.t. function ϕ. The right-hand side of (3) can be reduced to $-\gamma h(x)$ with $\gamma>0$ for simplicity.

[1]An extended $\mathcal{K}_\infty$ function Γ is a continuous strictly increasing function that maps $\mathbb{R}$ to $\mathbb{R}$ with $\Gamma(0)=0$.

Additionally, given the CBF h, consider the set

$$K_{\text{cbf}}(x) = \{u\in\mathcal{U} \mid L_f h(x) + L_g h(x)u \geq -\Gamma(h(x))\}. \tag{4}$$

We can prove that the control $\pi(x)\in K_{\text{cbf}}(x)$ renders C forward-invariant if the initial condition $x_0\in C$ [12].

III. Differentiable Optimization

In this section, we extend the results of [11] and show that the solution of a broader class of optimization problems is continuously differentiable w.r.t. the underlying parameters. First, we introduce the definition of a scaling function.

Definition 2 (Scaling functions): We say that a class $\mathcal{C}^2$ function $\mathcal{F}_A:\mathbb{R}^n\to\mathbb{R}$ is a scaling function for a non-empty closed set $A\subset\mathbb{R}^n$ if

1) $A = \{z\in\mathbb{R}^n \mid \mathcal{F}_A(z)\leq 1\}$;
2) $\mathcal{F}_A(z)<\infty$ for all $z\in\mathbb{R}^n$.

The above definition is slightly modified from that in [11] as we generalize the domain of $\mathcal{F}_A$ to n-dimensional space and allow the range of $\mathcal{F}_A$ to be negative. The additional condition 2) guarantees that it is always possible to include a point $z\in\mathbb{R}^n$ in a sublevel set of $\mathcal{F}_A$. It should be noted that the scaling function does not always exist for a given set. However, when a scaling function exists, the convexity of the scaling function implies the convexity of the associated set[2].

Some sets come with a natural representation of scaling functions. For a 2D (or 3D) ball of radius r, we can write $\mathcal{F}_A(z) = (z-z_0)^\top P(z-z_0)$ where z_0 is the center of the ball and $P = I/r^2$. Similarly, we can find the scaling function of an ellipse (or an ellipsoid) by choosing P properly.

In the following, we are interested in the case where the scaling functions are convex. The next lemma results from the first-order condition for a differentiable convex function.

Lemma 1: Given a convex scaling function $\mathcal{F}_A$, for any z such that $\mathcal{F}_A(z)>1$, $\frac{\partial\mathcal{F}_A}{\partial z}(z)\neq 0$.

Proof: As $\mathcal{F}_A$ is convex and belongs to class $\mathcal{C}^2$, we have

$$\mathcal{F}_A(z') \geq \mathcal{F}_A(z) + \frac{\partial\mathcal{F}_A}{\partial z}(z)(z'-z),\quad \forall z'\in\mathbb{R}^n.$$

If $\frac{\partial\mathcal{F}_A}{\partial z}(z)=0$, then $\mathcal{F}_A(z')\geq\mathcal{F}_A(z)$ for any $z'\in\mathbb{R}^n$, i.e., z is a global minimizer of $\mathcal{F}_A$. However, this is impossible because $A=\{z\in\mathbb{R}^n \mid \mathcal{F}_A(z)\leq 1\}$ is non-empty. Therefore, $\frac{\partial\mathcal{F}_A}{\partial z}(z)\neq 0$ must hold. □

Let $\mathcal{F}_A$ and $\mathcal{F}_B$ be two convex scaling functions associated with A and B. Assume that the two sets A and B do not intersect, i.e., $A\cap B=\varnothing$. The minimal scaling factor such that the scaled versions of A and B intersect can be found via the following optimization problem [10], [11]:

$$\begin{aligned}\min_{z,\alpha}\quad & \alpha\\ \text{s.t.}\quad & \mathcal{F}_A(z)\leq\alpha,\quad \mathcal{F}_B(z)\leq\alpha.\end{aligned} \tag{5}$$

Problem (5) is strictly feasible because $\mathcal{F}_A(z),\mathcal{F}_B(z)<\infty$ for any $z\in\mathbb{R}^n$ (by Definition 2), i.e., we can increase α until the

[2]Pick $z_1,z_2\in A$, and by Definition 2, $\mathcal{F}_A(z_1)\leq 1$ and $\mathcal{F}_A(z_2)\leq 1$. Take $0\leq\eta\leq 1$, and by conveixty of $\mathcal{F}_A$, $\mathcal{F}_A(\eta z_1+(1-\eta)z_2)\leq\eta\mathcal{F}_A(z_1)+(1-\eta)\mathcal{F}_A(z_2)\leq 1$, which means that $\eta z_1+(1-\eta)z_2\in A$ and hence the convexity of A.

scaled versions of A and B intersect. The Lagrangian function of the above problem is

$$L(\alpha, z, \lambda_A, \lambda_B) = \alpha + \lambda_A(\mathcal{F}_A(z) - \alpha) + \lambda_B(\mathcal{F}_B(z) - \alpha). \tag{6}$$

Since the problem is convex and strictly feasible (i.e., Slater's condition holds), the KKT conditions are necessary. The optimal primal (α^*, z^*) and dual $(\lambda_A^*, \lambda_B^*)$ variables satisfy the following KKT conditions:

$$\lambda_A^* + \lambda_B^* = 1, \tag{7a}$$

$$\lambda_A^* \frac{\partial \mathcal{F}_A}{\partial z}(z^*) + \lambda_B^* \frac{\partial \mathcal{F}_B}{\partial z}(z^*) = 0, \tag{7b}$$

$$\lambda_A^*(\mathcal{F}_A(z^*) - \alpha^*) = 0, \tag{7c}$$

$$\lambda_B^*(\mathcal{F}_B(z^*) - \alpha^*) = 0, \tag{7d}$$

$$\lambda_A^*, \lambda_B^* \geq 0. \tag{7e}$$

Lemma 2: If $\mathcal{F}_A$ and $\mathcal{F}_B$ are both convex and $A \cap B = \varnothing$, then $\lambda_A^*, \lambda_B^* > 0$, $\mathcal{F}_A(z^*) = \mathcal{F}_B(z^*) = \alpha^*$, and $\frac{\partial \mathcal{F}_A}{\partial z}(z^*) \neq \frac{\partial \mathcal{F}_B}{\partial z}(z^*)$.

Proof: First, it is easy to see $\alpha^* > 1$ as $A \cap B = \varnothing$ and the problem is feasible. Next, we prove $\lambda_A^*, \lambda_B^* > 0$. If $\lambda_A^* = 0$, we have $\lambda_B^* = 1$ by (7a). Further, from (7b) and (7d), we get $\frac{\partial \mathcal{F}_B}{\partial z}(z^*) = 0$ and $\mathcal{F}_B(z^*) = \alpha^* > 1$, and this contradicts Lemma 1. Therefore, $\lambda_A^* \neq 0$, and similarly $\lambda_B^* \neq 0$. By (7e), $\lambda_A^*, \lambda_B^* > 0$. Following from $\lambda_A^*, \lambda_B^* > 0$, we have $\mathcal{F}_A(z^*) = \mathcal{F}_B(z^*) = \alpha^*$ by (7c) and (7d). Finally, we prove $\frac{\partial \mathcal{F}_A}{\partial z}(z^*) \neq \frac{\partial \mathcal{F}_B}{\partial z}(z^*)$. Assume $\frac{\partial \mathcal{F}_A}{\partial z}(z^*) = \frac{\partial \mathcal{F}_B}{\partial z}(z^*)$, then from (7a) and (7b), $\frac{\partial \mathcal{F}_A}{\partial z}(z^*) = \frac{\partial \mathcal{F}_B}{\partial z}(z^*) = 0$. This again contradicts Lemma 1. Hence, $\frac{\partial \mathcal{F}_A}{\partial z}(z^*) \neq \frac{\partial \mathcal{F}_B}{\partial z}(z^*)$. □

Assume that the scaling functions $\mathcal{F}_A$ and $\mathcal{F}_B$ are also functions of a parameter vector θ. Let α^* be the solution to problem (5). The following theorem establishes the continuous differentiability of α^* w.r.t. θ.

Theorem 1: Assume that $A \cap B = \varnothing$, and $\mathcal{F}_A$ and $\mathcal{F}_B$ are also $\mathcal{C}^2$ in θ. If $\mathcal{F}_A$ and $\mathcal{F}_B$ are both convex w.r.t. z and one of them is strictly convex w.r.t. z, then α^* is continuously differentiable w.r.t. θ.

Proof: Recall that $\lambda_A^* \frac{\partial \mathcal{F}_A}{\partial z}(z^*, \theta) + (1 - \lambda_A^*) \frac{\partial \mathcal{F}_B}{\partial z}(z^*, \theta) = 0$ and $\mathcal{F}_A(z^*, \theta) - \mathcal{F}_B(z^*, \theta) = 0$. Differentiating both equations w.r.t. θ gives

$$\underbrace{\begin{bmatrix} M & c \\ c^\top & 0 \end{bmatrix}}_{N} \begin{bmatrix} \frac{\partial z^*}{\partial \theta} \\ \frac{\partial \lambda_A^*}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} \tag{8}$$

where

$$M = \lambda_A^* \frac{\partial^2 \mathcal{F}_A}{\partial z^2}(z^*, \theta) + (1 - \lambda_A^*) \frac{\partial^2 \mathcal{F}_B}{\partial z^2}(z^*, \theta), \tag{9a}$$

$$c = \frac{\partial \mathcal{F}_A}{\partial z}(z^*, \theta)^\top - \frac{\partial \mathcal{F}_B}{\partial z}(z^*, \theta)^\top, \tag{9b}$$

$$\Omega_1 = -\lambda_A^* \frac{\partial^2 \mathcal{F}_A}{\partial \theta \partial z}(z^*, \theta) - (1 - \lambda_A^*) \frac{\partial^2 \mathcal{F}_A}{\partial \theta \partial z}(z^*, \theta), \tag{9c}$$

$$\Omega_2 = -\frac{\partial \mathcal{F}_A}{\partial \theta}(z^*, \theta) + \frac{\partial \mathcal{F}_B}{\partial \theta}(z^*, \theta). \tag{9d}$$

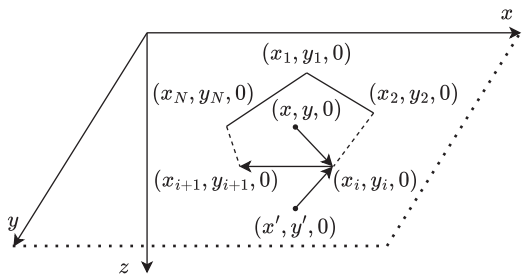


Fig. 1. 2D convex polygon extended into a 3D space.

Since $\mathcal{F}_A$ and $\mathcal{F}_B$ are both convex w.r.t. z with one of them being strictly convex, $\lambda_A^* > 0, 1 - \lambda_A^* = \lambda_B^* > 0$, M is positive definite. Furthermore, $c \neq 0$ using Lemma 2. Therefore, by Schur complement, the matrix N is invertible. Since M, c, Ω_1, and Ω_2 are continuous in θ, N^{-1} is continuous in θ and $\frac{\partial z^*}{\partial \theta}(\theta)$ is continuous in θ. Finally, differentiate $\mathcal{F}_A(z^*, \theta) - \alpha^* = 0$ w.r.t. θ and we see

$$\frac{\partial \alpha^*}{\partial \theta}(\theta) = \frac{\partial \mathcal{F}_A}{\partial z}(z^*, \theta) \frac{\partial z^*}{\partial \theta}(\theta) + \frac{\partial \mathcal{F}_A}{\partial \theta}(z^*, \theta) \tag{10}$$

is continuous in θ. □

Theorem 1 only states that given two scaling functions, satisfying the assumptions above will lead to the continuous differentiability of α^*. It does not guarantee the existence of such scaling functions for any convex set. A systematic construction method to derive scaling functions for 2D convex polygons will be presented in Section IV.

IV. Scaling Functions for 2D Convex Polygons

In this section, we derive scaling functions for 2D convex polygons. The main idea is to pad the polygon into a strictly convex set with tunable accuracy.

A. Inequalities Describing a 2D Convex Polygon

In this work, we use normalized image points[3]. In the 2D image plane, a convex polygon S with N edges can be delimited by N linear inequalities corresponding to each of its edges, i.e., $S = \{z \in \mathbb{R}^2 \mid Fz + b \leq 0\}$ where $F \in \mathbb{R}^{n \times 2}$ and $b \in \mathbb{R}^n$. In the following, given the vertex coordinates $\theta = [x_1, y_1, \ldots, x_N, y_N] \in \mathbb{R}^{2N}$ of S, we present a systematic method to find $F(\theta)$ and $b(\theta)$.

A convex polygon S in the image plane can be extended into a 3D space by appending 0 as the third element (see Fig. 1). Let $v_i = [x_i, y_i, 0]^\top$ be the i-th vertex of S ordered in the clockwise direction. In addition, let $s = [x, y, 0]^\top$ be a point inside S and $s' = [x', y', 0]^\top$ a point outside S. Using the right-hand rule, we see that the cross products $(v_i - s) \times (v_{i+1} - v_i)$ and $(v_i - s') \times (v_{i+1} - v_i)$ have different signs in the third component. By requiring the third component of $(v_i - s) \times (v_{i+1} - v_i)$ to

[3]In the pinhole camera model, the unitless normalized image point (x, y) associated with the image point (u, v) in pixel units is $(x, y) = ((u - c_x)/f_x, (v - c_y)/f_y)$, where (f_x, f_y) are the focal lengths and (c_x, c_y) are the principal point offsets.

be positive, the edge linking v_i and v_{i+1} leads to the following linear inequality:

$$(y_{i+1}-y_i)x+(x_i-x_{i+1})y+x_{i+1}y_i-x_iy_{i+1}\leq 0. \quad (11)$$

Putting all the inequalities in a compact form, we get

$$\underbrace{\begin{bmatrix} y_2-y_1 & x_1-x_2 \\ \vdots & \vdots \\ y_1-y_N & x_N-x_1 \end{bmatrix}}_{F(\theta)} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{z\in\mathbb{R}^2} + \underbrace{\begin{bmatrix} x_2y_1-x_1y_2 \\ \vdots \\ x_1y_N-x_Ny_1 \end{bmatrix}}_{b(\theta)} \leq 0. \quad (12)$$

In addition, $\frac{\partial F}{\partial \theta}$ and $\frac{\partial b}{\partial \theta}$ exist and are continuous in θ.

Remark 1: The content presented in Section IV-A can be taken as a specific instance within the wider scope of characterizing the convex hull of a simple polygon [13].

B. Padding the Polygon Into a Strictly Convex Set

The next step is to convert the 2D convex polygon $S=\{z\in\mathbb{R}^2 \mid Fz+b\leq 0\}$ into a strictly convex set, where $F=F(\theta)$ and $b=b(\theta)$ for short. Let the i-th row of F and b be F_i and b_i, respectively. Consider the set $S_s=\{z\in\mathbb{R}^2 \mid \sum_{i=1}^N \exp[\kappa(F_iz+b_i)]\leq N\}$ with $\kappa>0$. It is straightforward to see that $S\subset S_s$ as $\exp(\kappa t)\in(0,1]$ for $t\in(-\infty,0]$ with $\kappa>0$. Define

$$\mathcal{F}_{S_s}(z)=\frac{1}{N}\sum_{i=1}^{N}\exp[\kappa(F_iz+b_i)] \quad (13)$$

as a scaling function for S_s and we can see $S_s=\{z\in\mathbb{R}^2 \mid \mathcal{F}_{S_s}(z)\leq 1\}$ by the definition of S_s. Furthermore,

$$\frac{\partial^2\mathcal{F}_{S_s}}{\partial z^2}(z)=\frac{\kappa^2}{N}\sum_{i=1}^{N}\underbrace{\exp[\kappa(F_iz+b_i)]}_{>0}F_i^TF_i>0. \quad (14)$$

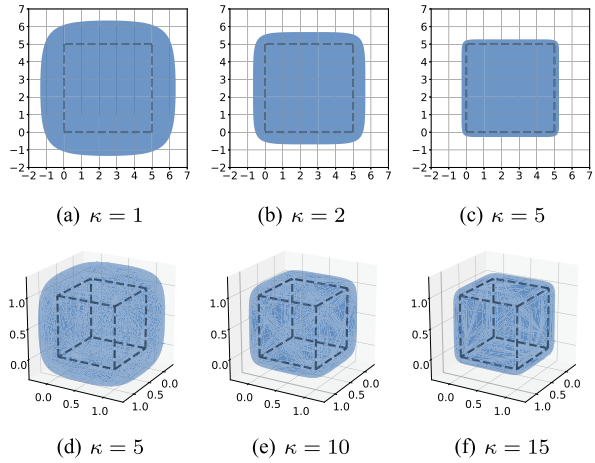
This is because the edges of a convex polygon cannot be all parallel, implying that there must exist two linearly independent rows in F. Hence, $\frac{\partial^2\mathcal{F}_{S_s}}{\partial z^2}(z)>0$, i.e., $\mathcal{F}_{S_s}$ is strictly convex w.r.t. z.

We remark that $\mathcal{F}_{S_s}$ comes with a tunable parameter κ that controls the accuracy of the padded shape S_s. As illustrated in Fig. 2, increasing κ will lead to a more accurate region (in blue) compared with the original shape. This method can also be applied to 3D convex polytopes as shown in Fig. 2 using the inequalities describing each face of the polytope.

Remark 2: When $\mathcal{F}_A=\mathcal{F}_{S_s}$ and $\mathcal{F}_B=\mathcal{F}_{S_s'}$, optimization problem (5) is equivalent to

$$\begin{aligned} \min_{z,\alpha'} \quad & \exp(\alpha') \\ \text{s.t.} \quad & \mathcal{F}_{S_s}(z)\leq\exp(\alpha'),\quad \mathcal{F}_{S_s'}(z)\leq\exp(\alpha'), \end{aligned} \quad (15)$$

with $\alpha=\exp(\alpha')$ as $\alpha>1$ if $S_s\cap S_s'=\varnothing$. The constraint $\mathcal{F}_{S_s}(z)\leq\exp(\alpha')$ (same for $\mathcal{F}_{S_s'}(z)\leq\exp(\alpha')$) can be further developed into a sum of exponential cones and this can be efficiently solved by solvers such as SCS (Splitting Conic Solver) and ECOS (Embedded Conic Solver).



(a) $\kappa=1$ (b) $\kappa=2$ (c) $\kappa=5$ (d) $\kappa=5$ (e) $\kappa=10$ (f) $\kappa=15$

Fig. 2. (a)–(c): Padded squares with $\kappa=1$, 2, and 5. (d)–(f): Padded cubes with $\kappa=5$, 10, and 15.

V. Motion Field Equations and Motion Observer

In this section, we recall some results on perspective projection and incorporate a motion observer into the motion field equations. Let there be M feature points and denote by $p_i=[x_i,y_i]^\top$ and Z_i the normalized image coordinates and depth in the camera frame of the i-th point, respectively. For the i-th point, standard calculation (e.g., [14]) gives the following motion field equation:

$$\dot{p}_i=L_i\begin{bmatrix} {}^cV_c-{}^cV_{i,w} \\ {}^c\Omega_c \end{bmatrix} \quad (16)$$

where ${}^cV_c\in\mathbb{R}^3$ and ${}^c\Omega_c\in\mathbb{R}^3$ are the translational and angular velocities of the camera expressed in the camera frame, ${}^cV_{i,w}$ is the velocity of the point expressed in the camera frame, and

$$L_i=\begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_iy_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1+y_i^2 & -x_iy_i & -x_i \end{bmatrix} \quad (17)$$

is the so-called image Jacobian or interaction matrix. The first three columns and the last three columns of L_i are noted as $L_{i,v}$ and $L_{i,\omega}$, respectively.

Our aim is to estimate $\dot{p}_i$ of M moving points with unknown motion profiles. In (16), L_i can be found using (17) and depth data. ${}^cV_c$ and ${}^c\Omega_c$ are our control inputs, but ${}^cV_{i,w}$ may not be readily available. This corresponds to the contribution of the moving speed of the point itself. For this purpose, we introduce the following motion observer (adapted from the disturbance observer in [15]). Stacking (16) for all M points, we obtain

$$\dot{p}=Lu+d \quad (18)$$

where $p=[p_1^\top,\ldots,p_M^\top]^\top\in\mathbb{R}^{2M}$, $u=[{}^cV_c{}^\top,{}^c\Omega_c{}^\top]^\top\in\mathbb{R}^6$, $L=[L_1^\top,\ldots,L_M^\top]^\top\in\mathbb{R}^{2M\times 6}$, and $d=[-(L_{1,v}{}^cV_{1,w})^\top,\ldots,-(L_{M,v}{}^cV_{M,w})^\top]^\top\in\mathbb{R}^{2M}$. In (18), we have summarized the $-L_{i,v}{}^cV_{i,w}$ terms into a disturbance vector d. Let the dynamics of the observer be

$$\hat{d}=Gp-\epsilon,\quad \dot{\epsilon}=G(Lu+\hat{d}) \quad (19a)$$

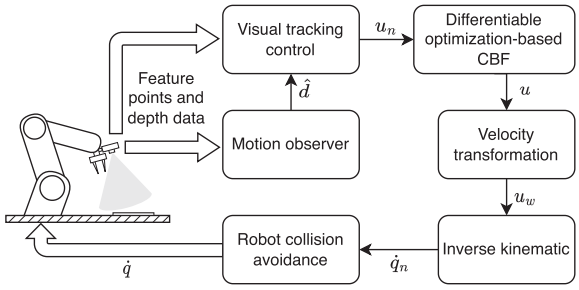


Fig. 3. Control process for occlusion-free VS.

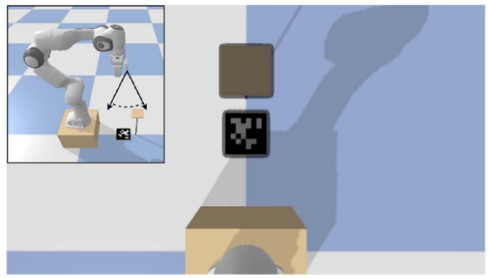


Fig. 4. Shaded areas (in black) represent the padded polygons for the obstacle (the square board) and the target (the board with an AprilTag). The overall scenario is shown in the upper-left figure.

where $G \in \mathbb{R}^{2M\times 2M}$ is the observer gain. Then, the estimation error $e = d - \hat{d}$ satisfies

$$\dot{e} = -Ge + \dot{d}. \tag{20}$$

Assuming that the feature points move *slowly* with a bounded variation, we establish the boundedness of the estimation error of the proposed motion observer.

Proposition 1: If $G > 0$ is diagonalizable and $\sup_{t\geq 0} \|\dot{d}\| < \infty$, then there exist $\epsilon_1, \epsilon_2, \sigma > 0$ such that $\|e(t)\| \leq \epsilon_1 e^{-\sigma t} + \epsilon_2(1 - e^{-\sigma t})$.

Proof: The solution to (20) is $e(t) = e^{-Gt}e_0 + \int_0^t e^{-G(t-\tau)}\dot{d}(\tau)d\tau$. Since $G > 0$ is diagonalizable, we have $G = P\Lambda P^{-1}$ with Λ diagonal. Then, $e^{-Gt} = Pe^{-\Lambda t}P^{-1}$ and $\|e^{-Gt}\| \leq \|P\|\|P^{-1}\|\|e^{-\Lambda t}\| \leq c_0 e^{-\sigma t}$ where $c_0 = \|P\|\|P^{-1}\|$ and σ is the smallest eigenvalue of G. Therefore,

$$\begin{aligned}\|e(t)\| &\leq c_0\|e_0\|e^{-\sigma t} + c_0 \sup_{t\geq 0}\|\dot{d}\| \int_0^t e^{-\sigma(t-\tau)}d\tau \\ &= \epsilon_1 e^{-\sigma t} + \epsilon_2(1 - e^{-\sigma t})\end{aligned}$$

with $\epsilon_1 = c_0\|e_0\|$ and $\epsilon_2 = c_0 \sup_{t\geq 0}\|\dot{d}\|/\sigma$. □

Remark 3: Typically, requiring $\sup_{t\geq 0}\|\dot{d}\| < \infty$ is equivalent to requiring $\|p_i\|, \|\dot{p}_i\|, \|{}^cV_{i,w}\|, \|{}^c\dot{V}_{i,w}\| < \infty$ and $Z_i > 0$ for all the feature points and for all $t \geq 0$. We believe these are mild conditions for practical implementation.

VI. Camera Control and Occlusion Avoidance

In this section, we introduce the different components of the camera control process (see Fig. 3).

A. Visual Tracking Control

In this letter, we track a target with features provided by an AprilTag [16]. The corners of the AprilTag are selected as feature points, and given the desired coordinates of the corners p^*, the nominal control law is

$$u_n = -L^+(\lambda(p - p^*) + \hat{d}), \tag{21}$$

where $\lambda > 0$, $\hat{d}$ is given by the motion observer, and L^+ is an approximated inverse of L chosen by the user (some possible choices discussed below).

If $M = 1, 2, 3$ and L has linearly independent rows, we can use $L^+ = L^\dagger$ (or simply L^{-1} when $M = 3$). In this case, the ultimate boundedness of the tracking error $p - p^*$ can be proved as in [17, Theorem 2] because the estimation error $e = d - \hat{d}$ is also bounded (Proposition 1). In this work, we choose to track all four corners of the AprilTag, and a damped pseudoinverse $L^+ = L^\top(LL^\top + \rho^2 I)^{-1}$ is used to avoid singularities in the simulation studies. For the real robot experiment, we use $L^+ = (L + L^*)^\dagger/2$, where L^* is the value of L when $p = p^*$. This special choice of L^+ has proven to provide good performance in practice [14].

B. Control Barrier Function

In this work, we focus on scenarios where the projections of the target and the obstacle in the image are both convex polygons, which can cover many user scenarios. For non-convex projections, we suggest finding their convex hulls [13], if this approach is feasible and not overly restrictive.

Let S and S' be the polygonal projection of the target and the obstacle in the image plane, respectively, and denote by p and p' the coordinates of their vertices, respectively. Given the p and p', one can construct the scaling functions $\mathcal{F}_{S_s}$ and $\mathcal{F}_{S'_s}$ as in Section IV. In addition, $\mathcal{F}_{S_s}$ and $\mathcal{F}_{S'_s}$ will have p and p' as their parameters, respectively. An example is shown in Fig. 4 where we visualize the padded shapes S_s and S'_s w.r.t. their original shape. We assume the target has an unknown motion profile and the obstacle is fixed in the world frame. Recall that the motion field equation for p is

$$\dot{p} = Lu + \hat{d} + e \tag{22}$$

where $e = d - \hat{d}$ is the estimation error of the motion observer. As the obstacle is fixed in the world frame, the motion field equation for p' is

$$\dot{p}' = L'u. \tag{23}$$

Both L and L' can be calculated by using the projection equation and depth data. Also, from (22) and (23), we can see the velocities of p and p' share the same control $u = [{}^cV_c{}^\top, {}^c\Omega_c{}^\top]^\top$ representing the camera speeds.

To avoid occlusion while tracking the moving target, we construct the following CBF

$$h(p, p') = \alpha^*(p, p') - \alpha_0 \tag{24}$$

where $\alpha^*(p, p')$ is the solution to the optimization problem (5) with $\mathcal{F}_A = \mathcal{F}_{S_s}$ and $\mathcal{F}_B = \mathcal{F}_{S'_s}$, and $\alpha_0 > 1$ is a user-chosen parameter. If the control u is designed such that $h(p, p') > 0$

holds forward in time, then $\alpha^*(p,p') > \alpha_0 > 1$. This means $S_s \cap S'_s = \varnothing$ and thus $S \cap S' = \varnothing$. In other words, the occlusion is avoided.

The continuous differentiability of h w.r.t. p and p' can be seen by Theorem 1 and the construction of $\mathcal{F}_{S_s}$ and $\mathcal{F}_{S'_s}$. The next theorem proves that $\nabla_{p,p'} h \neq 0$ for $h(p,p') = 0$.

Theorem 2: For p, p' such that $h(p,p') = 0$, $\nabla_{p,p'} h = [\frac{\partial h}{\partial p}, \frac{\partial h}{\partial p'}]^\top \neq 0$.

Proof: When $h(p,p') = 0$, we have $\alpha^*(p,p') = \alpha_0 > 1$ and thus $S_s \cap S'_s = \varnothing$. If $\nabla_{p,p'} h = 0$, then no infinitesimally small change in p or p' exists that can change α^* as $\nabla_{p,p'} \alpha^* = \nabla_{p,p'} h$. This is impossible because when $S_s \cap S'_s = \varnothing$, there always exists an infinitesimal translation of p and p' that will change the value of α^*. Hence, $\nabla_{p,p'} h \neq 0$ must hold.

Consider the following set with $\gamma > 0$ and $\mathcal{U} = \mathbb{R}^6$

$$K_{\text{cbf}}(p,p') = \Big\{ u \in \mathcal{U} \mid \frac{\partial h}{\partial p}(Lu + \hat{d}) + \frac{\partial h}{\partial p'} L' u \geq -\gamma h + E \Big\}. \tag{25}$$

Theorem 3: Assume that the conditions of Proposition 1 hold and $h(p_0, p'_0) > 0$ for the initial conditions p_0 and p'_0. If $\|\frac{\partial h}{\partial p}\| \leq k_h$, $E \geq k_h \max(\epsilon_1, \epsilon_2)$, and the control is chosen such that $u \in K_{\text{cbf}}(p,p')$, then $h(p,p') > 0$ holds for $t \geq 0$.

Proof: By (22) and (23), the time derivative of h is

$$\dot{h} = \frac{\partial h}{\partial p}(Lu + \hat{d} + e) + \frac{\partial h}{\partial p'} L' u.$$

Since $u \in K_{\text{cbf}}(p,p')$, we have

$$\dot{h} \geq \frac{\partial h}{\partial p} e - \gamma h + E \geq -\gamma h - \left\| \frac{\partial h}{\partial p} \right\| \|e\| + E.$$

As $\|\frac{\partial h}{\partial p}\| \leq k_h$, $\|e\| \leq \max(\epsilon_1, \epsilon_2)$ (by Proposition 1), and $E \geq k_h \max(\epsilon_1, \epsilon_2)$, we have $\dot{h} \geq -\gamma h$. Since $h(p_0, p'_0) > 0$ by assumption, we have $h(p,p') > 0$ for $t \geq 0$.

In practice, (25) is converted to a constraint of quadratic programming (QP). The so-called CBF-QP for this work is

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & \|u - u_n\|^2 \\ \text{s.t.} \quad & \frac{\partial h}{\partial p}(Lu + \hat{d}) + \frac{\partial h}{\partial p'} L' u \geq -\gamma h + E \end{aligned} \tag{26}$$

where u_n is the nominal controller derived in (21).

C. Inverse Kinematic and Robot Collision Avoidance

In this work, we use a FR3 robotic arm with the eye-in-hand configuration as our robotic platform. Given the u, the camera speeds expressed in the camera frame, we can convert u to the world frame[4] and denote by u_w the camera speeds expressed in the world frame. Let $q \in \mathbb{R}^{n_q}$ be the joint angles of the FR3 robotic arm. We have the following differential kinematics equation

$$u_w = J(q)\dot{q} \tag{27}$$

[4]This can be done by using known results on translating the velocities between a rotating frame and an inertial frame.

where $J(q)$ is the camera geometric Jacobian. Denote by $q_{i,l}$ and $q_{i,u}$ the lower and upper bounds of the i-th joint, respectively, and let $\bar{q}_i = (q_{i,l} + q_{i,u})/2$. As a secondary objective, we encourage each joint angle to stay in the middle of its limits. Defining $O(q) = -\frac{1}{n_q} \sum_{i=1}^{n_q} (q_i - \bar{q}_i)^2 / (q_{i,u} - q_{i,l})^2$, the nominal joint velocities are given by

$$\dot{q}_n = J(q)^\dagger u_w + (I - J(q)^\dagger J(q)) \frac{\partial O}{\partial q}(q)^\top. \tag{28}$$

To guarantee that the robot does not physically collide with the obstacle, we incorporate the obstacle-avoidance CBF (OA-CBF) in [11]. As shown in Figs. 5 and 6, the robot links of interest are wrapped in blue ellipsoids (except a sphere for the end-effector) that serve as strongly convex bounding shapes. Then, OA-CBF performs a QP that returns the safe joint velocities $\dot{q}$ with a minimal ℓ_2 error from $\dot{q}_n$.

VII. Experimental Studies

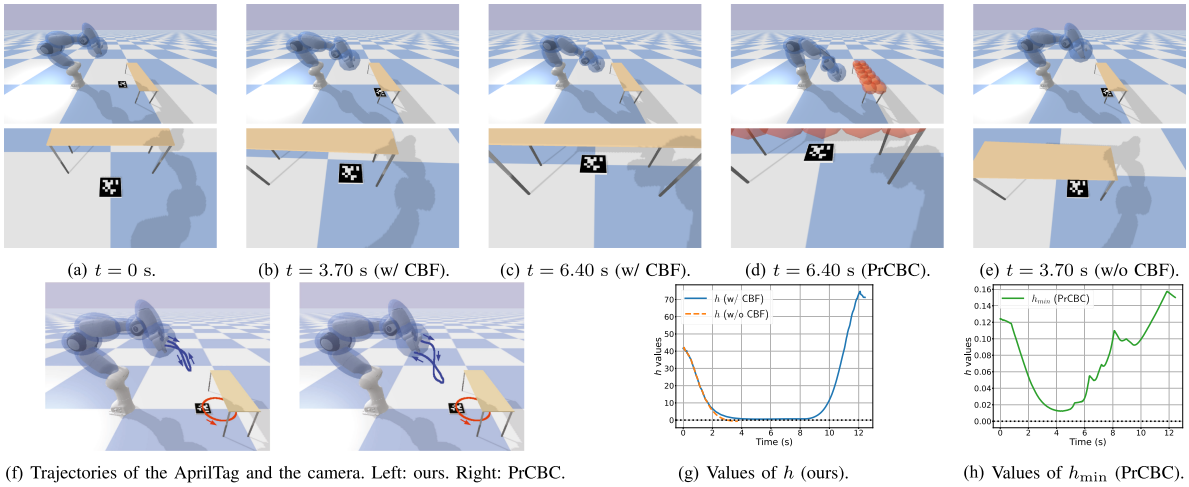
The FR3 robotic arm platform with the eye-in-hand configuration (see Fig. 4) is utilized in our experimental studies. We employ cvxpylayers [9] to solve (5) and compute the partial differentials w.r.t. the parameters. Additionally, we use Proxsuite [18] to solve QPs and Pinocchio [19] to compute the pose of different frames of the robot. Throughout our experiments, we know the geometry of the environment and the objects appearing in the environment. For our experiments, one obstacle is fixed in the world frame with a known pose while the target (a small board with an AprilTag) has an unknown motion profile satisfying the conditions of Proposition 1. The video of the experiments is available at https://youtu.be/su0RPjcBwOc.

A. Simulation 1: Target Going Under the Obstacle

Consider the scenario (in Fig. 5) where the target goes under the rectangular-shaped obstacle ($0.20\ \text{m} \times 0.60\ \text{m}$) at a height of $0.25\ \text{m}$. The target follows a circular trajectory with a radius of $0.15\ \text{m}$ and an angular velocity of $0.50\ \text{rad/s}$ (see Fig. 5(f)). While the target is maneuvered to go under the obstacle in the simulation, the FR3 robotic arm should keep the target in the image plane with least occlusion.

The initial pose of the FR3 robotic arm and the camera image are shown in Fig. 5(a). If the CBF-QP is not used, the target will get occluded at $t = 3.70\ \text{s}$ (see Fig. 5(e)). If the CBF-QP is used, the FR3 robotic arm will exhibit proper motions (i.e., a tilting behavior) to avoid occlusion (see Figs. 5(b) and (c)) and the overall camera trajectory is visualized in Fig. 5(f). Finally, the values of the h function defined in (24) are shown in Fig. 5(g). As expected, $h \geq 0$ holds during the simulation; therefore, occlusion is avoided.

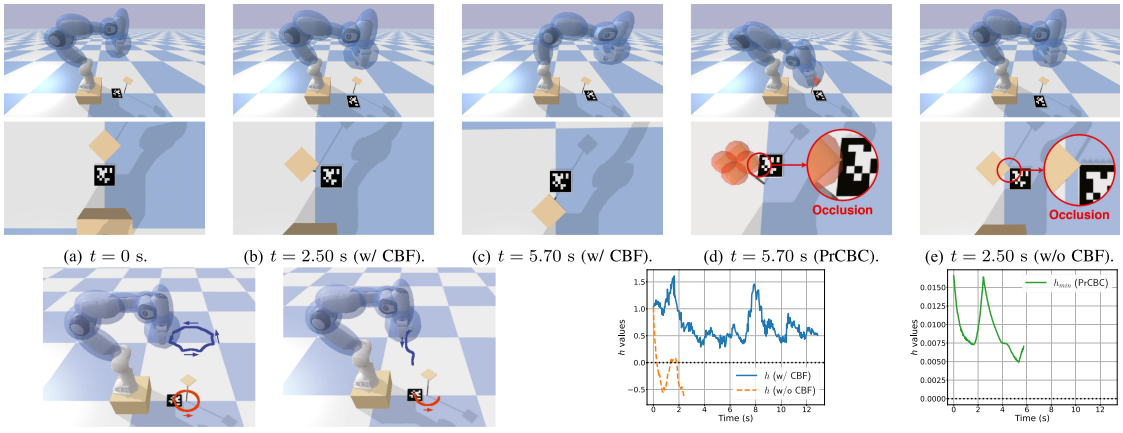
We compare our method with the state-of-the-art occlusion avoidance method called PrCBC (Probabilistic Control Barrier Certificates) in [6]. PrCBC models the obstacles with rigid spheres and constructs CBFs to guarantee that the feature points do not enter the projection of these spheres in the image space. As shown in Fig. 5(d), the rectangular-shaped obstacle is contained within $n_s = 12$ rigid spheres of radius $0.074\ \text{m}$. The four ($n_f = 4$) corners of the AprilTag are selected as feature points,



(a) $t = 0$ s. (b) $t = 3.70$ s (w/ CBF). (c) $t = 6.40$ s (w/ CBF). (d) $t = 6.40$ s (PrCBC). (e) $t = 3.70$ s (w/o CBF).

(f) Trajectories of the AprilTag and the camera. Left: ours. Right: PrCBC. (g) Values of h (ours). (h) Values of $h_{\min}$ (PrCBC).

Fig. 5. Target going under the obstacle. (a): Initial robot pose and camera image. (b) and (c): Robot poses and camera images (w/ CBF). (d): Robot pose and camera image (PrCBC). (e): Robot pose and camera image (w/o CBF) when the occlusion happens. (f): Trajectories of the center of the target (in red) and the camera (in blue). (g): Values of h w/ and w/o CBF. (h): Values of $h_{\min}$ for PrCBC.



(a) $t = 0$ s. (b) $t = 2.50$ s (w/ CBF). (c) $t = 5.70$ s (w/ CBF). (d) $t = 5.70$ s (PrCBC). (e) $t = 2.50$ s (w/o CBF).

(f) Trajectories of the AprilTag and the camera. Left: ours. Right: PrCBC. (g) Values of h (ours). (h) Values of $h_{\min}$ (PrCBC).

Fig. 6. Target moving around the obstacle. (a): Initial robot pose and camera image. (b) and (c): Robot poses and camera images (w/ CBF). (d): Robot pose and camera image when the occlusion happens (PrCBC). (e): Robot pose and camera image (w/o CBF) when the occlusion happens. (f): Trajectories of the target (in red) and the camera (in blue). (g): Values of h w/ and w/o CBF. (h): Values of $h_{\min}$ for PrCBC.

and a total number of $n_s n_f = 48$ CBFs are employed by PrCBC. We visualize the minimum of all the CBFs in Fig. 5(h). Due to the conservativeness of obstacle modeling, PrCBC generates over-reactive tilting behavior (see Fig. 5(f)). We numerically calculate the cost of control (in terms of joint velocities) $\int_0^T \dot{q}^\top \dot{q} dt$, and the cost of our method is 0.71 $\mathrm{rad}^2/\mathrm{s}$ while that of PrCBC is 2.55 $\mathrm{rad}^2/\mathrm{s}$, which is 3.59 times higher than our method.

B. Simulation 2: Target Moving Around the Obstacle

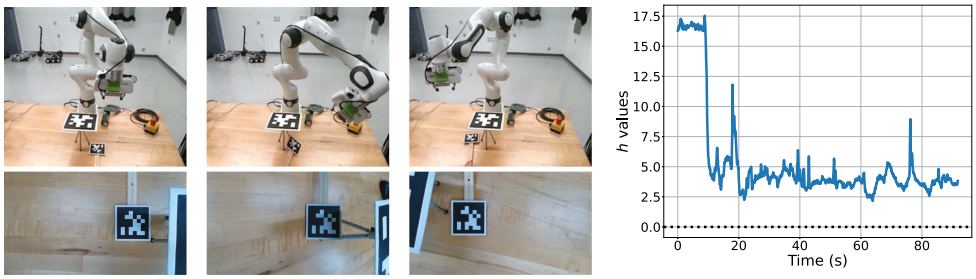
In the second example, we consider a scenario (in Fig. 6) where the target moves around a square-shaped obstacle ($0.10\,\mathrm{m} \times 0.10\,\mathrm{m}$) at a height of 0.3 m. The target again follows a circular trajectory with a radius of 0.12 m and an angular velocity of 0.50 rad/s (see Fig. 6(f)).

Without the CBF, the upper-left corner of the target will get occluded at $t = 2.50$ s (see Fig. 6(e)). When the CBF-QP is active, the FR3 robotic arm will generate a tilting behavior when h is near zero to avoid occlusions (see Figs. 6(b) and (c)), and the overall camera trajectory is visualized in Fig. 6(f). The values of h with and without CBF are shown in Fig. 6(g).

In this comparison with PrCBC, the four corners of the AprilTag are selected as the feature points and the obstacle is covered with four rigid spheres of radius 0.037 m (see Fig. 6(d)). Although all 16 employed CBFs have positive values (see Fig. 6(h)), occlusion still happens (see Fig. 6(d)). This shows that merely having visible feature points does not assure that the target's visual appearance remains unobstructed. Conversely, our approach takes into account the convex shapes of both the target and the obstacle, thereby effectively ensuring avoidance of occlusion in this scenario.

C. Experiments on a 7-DOF Franka Arm

For the real robot experiment, we test our proposed approach on a FR3 robotic arm platform with the computational engine



(a) Snapshots 1. (b) Snapshots 2. (c) Snapshots 3. (d) Values of h.

Fig. 7. Real robot experiments. (a)–(c): Snapshots of the external view and robot camera view. (d): Values of h.

being a PC with 32 GB RAM and an Intel Core i7 11700 processor. An Extended Kalman Filter as in [20] is implemented to denoise the depth signal and detection results from the perception pipeline. This work employs joint velocity control, and the control update rate is 20 Hz. The robot carries a D435i camera that outputs color images of size 1280×720 at 30 Hz. The target and obstacle are of size $8.6 \times 8.6\,\mathrm{cm}$ and $17.2 \times 17.2\,\mathrm{cm}$, respectively. The obstacle is mounted at an approximate height of 16 cm from the table. See Fig. 7 for snapshots of the experiment and evolution of the h function.

VIII. Discussions and Conclusion

This work addresses the challenge of tracking a moving object in the FoV while avoiding occlusion. A notable strength of our work is its tunable accuracy. Additionally, as our method pads convex polygons into strictly convex shapes using scaling functions without changing the number of constraints or decision variables, the complexity of our method does not increase as long as the number of obstacles and targets in the environment remains unchanged. This feature is advantageous, especially with large obstacles or targets.

Our approach, similar to other CBF-based methods, depends on initial conditions (e.g., initial camera pose) and the nominal control (e.g., desired camera pose). Our method ensures that the object stays within the FoV given some reasonable assumptions (e.g., the object velocity is not beyond the tracking capability of the arm, the object does not move behind obstacles outside the workspace of the manipulator given the geometry of the environments, etc.). Our findings indicate that imposing closer tracking distances increases the risk of losing the target from the FoV. We included some illustrative examples in the video for the readers to better understand these limitations.

In conclusion, our method, utilizing the minimal scaling factor, improves accuracy and complexity compared to the state-of-the-art method in [6]. Future research will focus on enhancing visual signal recovery and making the control design more robust to different tracking objectives.

References

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, U.K.: Springer, 2008.

[2] V. Cadenat, D. Folio, and A. D. Petiteville, "Comparison of two sequencing techniques to perform a vision-based navigation task in a cluttered environment," *Adv. Robot.*, vol. 26, no. 5/6, pp. 487–514, 2012.

[3] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 796–803, Apr. 2018.

[4] K. He et al., "Visibility maximization controller for robotic manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8479–8486, Jul. 2022.

[5] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3725–3732, Oct. 2018.

[6] Y. Zhang, Y. Yang, and W. Luo, "Occlusion-free image-based visual servoing using probabilistic control barrier certificates," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 4381–4387, 2023.

[7] T. Fujinami, J. Yamauchi, R. Funada, and M. Fujita, "A control barrier function approach for observer-based visually safe pursuit control with spherical obstacles," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 10799–10804, 2023.

[8] A. D. Petiteville, S. Durola, V. Cadenat, and M. Courdesses, "Management of visual signal loss during image based visual servoing," in *Proc. Eur. Control Conf.*, 2013, pp. 2305–2310.

[9] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019.

[10] K. Tracy, T. A. Howell, and Z. Manchester, "Differentiable collision detection for a set of convex primitives," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3663–3670.

[11] B. Dai, R. Khorrambakht, P. Krishnamurthy, V. Gonçalves, A. Tzes, and F. Khorrami, "Safe navigation and obstacle avoidance using differentiable optimization based control barrier functions," *IEEE Robot. Automat. Lett.*, vol. 8, no. 9, pp. 5376–5383, Sep. 2023.

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

[13] R. L. Graham and F. F. Yao, "Finding the convex hull of a simple polygon," *J. Algorithms*, vol. 4, no. 4, pp. 324–331, 1983.

[14] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Automat. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[15] A. Stotsky and I. Kolmanovsky, "Application of input estimation techniques to charge estimation and control in automotive engines," *Control Eng. Pract.*, vol. 10, no. 12, pp. 1371–1383, 2002.

[16] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.

[17] N. R. Gans, G. Hu, K. Nagarajan, and W. E. Dixon, "Keeping multiple moving targets in the field of view of a mobile camera," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 822–828, Aug. 2011.

[18] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "PROX-QP: Yet another quadratic programming solver for robotics and beyond," in *Proc. Robot.: Sci. Syst.*, 2022.

[19] J. Carpentier et al., "The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *Proc. IEEE/SICE Int. Symp. System Integration*, 2019, pp. 614–619.

[20] A. A. Oliva, E. Aertbeliën, J. D. Schutter, P. R. Giordano, and F. Chaumette, "Towards dynamic visual servoing for interaction control and moving targets," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 150–156.