Hierarchical Control for Occlusion-Free Visual Servoing of Robotic Manipulators

Yujie Wang, Tianxiao Ye, and Xiangru Xu

***Abstract*—Visual servoing leverages image features as feedback signals for robotic control, but existing methods generally lack rigorous formal guarantees for occlusion-free operation. This article presents a hierarchical optimization-based visual servoing framework for robotic manipulators that ensures strict occlusion-free operation in continuous time. The framework integrates a high-level model predictive control with a low-level control barrier function-based controller. Occlusion-avoidance constraints are reformulated into differentiable convex constraints via a vertex-based image representation and a duality-based optimization approach, enabling seamless incorporation into the model predictive control. The control barrier functions are constructed from the relaxed minimum distance between polytopes, with distance derivatives computed using the KKT conditions of the distance problem. The resulting framework simultaneously achieves target feature-point regulation and continuous-time enforcement of occlusion-avoidance constraints. Hardware experiments on a Franka Research 3 manipulator validate the real-time implementability and strict occlusion avoidance of the proposed approach.**

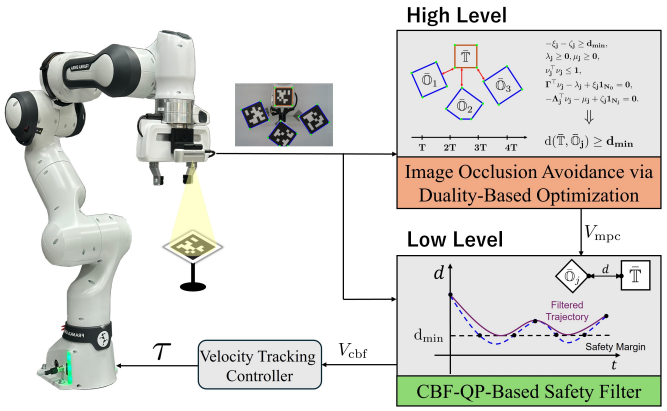


Figure 1: Architecture of the proposed hierarchical optimization-based visual servoing framework for robotic manipulators designed to ensure occlusion-free operation in continuous time.

I. Introduction

Vision-based control has emerged as a powerful paradigm for robotic systems, enabling interaction with dynamic environments through direct use of visual feedback. Various approaches have been developed and implemented across diverse applications [1], [2], [3], [4], [5], [6]. However, most existing approaches lack rigorous safety guarantees, which can result in performance degradation or even catastrophic failure in safety-critical applications. Several recent works have focused on developing vision-based controllers with formal safety guarantees [7], [8], [9]. For example, [7] employs a learned perception map that predicts a linear function of the state to design safe sets and robust controllers; however, the resulting safety regions are restricted to neighborhoods of the training data, and the smoothness assumptions are rather restrictive. The extension in [8] proposes state-dependent bounds for convolutional neural network-based state estimation and synthesizes robust controllers via linear matrix inequalities for spacecraft guidance, but the safe sets remain small, making the optimization problem difficult to solve. In [9], the perception model is approximated by piecewise affine functions, yet quantifying the relationship between the real and approximate models remains an open challenge.

Among vision-based control strategies, visual servoing, which uses image features as feedback signals for control, is a well-established and widely adopted framework with a rich history [10], [11], [12]. Its structured formulation facilitates the systematic development of rigorous guarantees. Visual servoing is typically divided into position-based visual servoing (PBVS), which reconstructs 3D pose information, and image-based visual servoing (IBVS), which regulates image features directly without explicit 3D reconstruction. Compared to PBVS, IBVS avoids solving nonlinear pose estimation problems online, offering greater computational efficiency and reduced sensitivity to calibration errors [10]. These advantages make IBVS particularly well-suited for applications demanding real-time performance or where precise calibration is challenging.

Despite significant progress, most IBVS algorithms are not designed for complex environments where targets may be occluded. In safety-critical applications such as surgical robotics [13], even brief visual occlusion is unacceptable. To address this challenge, several occlusion-free IBVS methods have been proposed using optimization-based control approaches [14], [15], [16], [17], [18], [19]. For example, [14] introduces a convex optimization-based controller that enforces occlusion-avoidance constraints in the image space for shared teleoperation of dual-arm robots, and [15] develops a model predictive control (MPC)-inspired online replanning strategy for quadcopters that avoid collision and occlusion by solving a sequence of nonlinear optimization problems using differential flatness and B-splines. More recently, control barrier function (CBF)-based methods have gained attention for occlusion avoidance because they provide strict safety guarantees and integrate seamlessly into convex quadratic

The authors are with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA. Email: {yujie.wang, tye46, xiangru.xu}@wisc.edu.

programs (QPs). For instance, [16] integrates probabilistic CBFs with MPC to account for feature point measurement noise; [17] leverages scaling functions and CBFs to formulate differentiable occlusion-avoidance constraints that preserve the geometry of occluding objects and reduce conservatism; and [18] introduces an occlusion-free controller for orthopedic surgical robots using visual cone models and discrete CBFs within an MPC framework. While CBF-based methods ensure real-time constraint satisfaction, they are inherently myopic and lack long-term optimality. In contrast, MPC-based approaches utilize predictive models to achieve better long-term performance but impose heavy online computational burdens and cannot guarantee continuous-time occlusion avoidance, since constraints are enforced only at discrete time steps.

In this article, we present a *hierarchical optimization-based* visual servoing framework for robotic manipulators that ensures *occlusion-free operation in continuous time*. The framework integrates two main components: a high-level MPC and a low-level CBF-based controller (see Figure 1). The high-level MPC regulates target feature points and generates a nominal reference input, while the low-level controller strictly enforces occlusion-avoidance constraints. This hierarchical design inherits the advantages of both MPC and CBFs. The main contributions of this work are summarized as follows.

- We reformulate the occlusion-avoidance constraints as differentiable convex constraints by leveraging a vertex-based image representation and a duality-based optimization approach. This reformulation enables the design of an MPC problem that regulates target feature points while enforcing occlusion avoidance in discrete time.
- We design a CBF-QP-based controller that strictly ensures the occlusion avoidance in continuous time while tracking the MPC reference input. The CBFs are constructed from the relaxed minimum distance between polytopes, with distance derivatives computed using the Karush–Kuhn–Tucker (KKT) conditions of the corresponding optimization problem, and their validity established via nonsmooth analysis.
- We experimentally validate the proposed hierarchical framework on a Franka Research 3 manipulator, demonstrating occlusion-free visual servoing and real-time implementability. Extensive comparisons highlight the effectiveness and advantages of the approach.

The remainder of this article is organized as follows. Section II introduces the preliminaries and problem statement. Section III presents the duality-based representation of image occlusion-avoidance constraints and the high-level MPC. Section IV describes the CBF-QP safety filter that strictly enforces occlusion-avoidance constraints. Section V reports experimental results on a Franka Research 3 manipulator equipped with an Intel RealSense camera. Finally, Section VI concludes the article.

Notation. For a positive integer n, denote $[n] = \{1, 2, \cdots, n\}$ and $[0, n] = \{0, 1, 2, \cdots, n\}$. For a vector $x \in \mathbb{R}^n$, let x_i denote its i-th entry, $\|x\|$ its 2-norm, $\|x\|_Q = \sqrt{x^\top Q x}$ where Q is a positive definite matrix, and $x \geq 0$ elementwise nonnegativity, i.e., $x_i \geq 0$ for any $i \in [n]$. Denote by I_n the $n \times n$ identity matrix and by $\mathbf{1}_n$ the n-dimensional column vector with all entries equal to 1. For a matrix $A \in \mathbb{R}^{n\times m}$, let A_{ij} denote its (i, j)-th entry and $\|A\|$ its Frobenius norm. Denote $\mathrm{diag}(a_1, a_2, \cdots, a_n) \in \mathbb{R}^{n\times n}$ as the diagonal matrix with entries $a_1, a_2, \cdots, a_n \in \mathbb{R}$, and $\mathrm{blkdiag}(A_1, A_2, \cdots, A_n)$ as the block-diagonal matrix with diagonal blocks $A_1, A_2, \cdots, A_n$, which need not be square. For a scalar function $h : \mathbb{R}^n \to \mathbb{R}$ with respect to $x \in \mathbb{R}^n$, the gradient $\frac{\partial h}{\partial x} \in \mathbb{R}^{1\times n}$ is considered a row vector. For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ with respect to $x \in \mathbb{R}^n$, $\frac{\partial f}{\partial x}$ denotes its Jacobian matrix, whose (i, j)-th entry is $\frac{\partial f_i}{\partial x_j}$. Given n points $p_1, p_2, \cdots, p_n \in \mathbb{R}^m$, their convex hull is defined as $\mathrm{conv}(p_1, \cdots, p_n) = \{\sum_{i=1}^n \lambda_i p_i : \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1\}$.

II. Preliminaries & Problem Statement

In this section, we introduce the necessary preliminaries (camera model, vertex-based representation of object images, and CBFs) and then formulate the problem investigated in this work.

A. Camera Model

Consider a point $P = [x, y, z]^\top \in \mathbb{R}^3$ in the camera coordinate frame, and let $\eta = [u_x\ u_y]^\top \in \mathbb{R}^2$ denote the image coordinates of its projection. By the perspective projection model [10], the *image coordinates* are given by

$$\eta = \frac{f}{z}\begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2, \tag{1}$$

where f is the camera focal length. By augmenting η with z, we define the *augmented state* as

$$\bar{\eta} = \begin{bmatrix} u_x \\ u_y \\ z \end{bmatrix} \in \mathbb{R}^3. \tag{2}$$

Its dynamics are given by [20]

$$\dot{\bar{\eta}} = J(\bar{\eta})\begin{bmatrix} v \\ \omega \end{bmatrix} - S(\bar{\eta})w, \tag{3}$$

where $v \in \mathbb{R}^3$ and $\omega \in \mathbb{R}^3$ denote the translational and angular velocities of the camera in the camera frame, respectively, $w \in \mathbb{R}^3$ is the *feature point velocity* in the camera frame, and

$$S(\bar{\eta}) = \begin{bmatrix} -\frac{f}{z} & 0 & \frac{u_x}{z} \\ 0 & -\frac{f}{z} & \frac{u_y}{z} \\ 0 & 0 & -1 \end{bmatrix}, \tag{4a}$$

$$J(\bar{\eta}) = \begin{bmatrix} -\frac{f}{z} & 0 & \frac{u_x}{z} & \frac{u_x u_y}{f} & -\frac{f^2+u_x^2}{f} & u_y \\ 0 & -\frac{f}{z} & \frac{u_y}{z} & \frac{f^2+u_y^2}{f} & -\frac{u_x u_y}{f} & -u_x \\ 0 & 0 & -1 & -\frac{u_y z}{f} & \frac{u_x z}{f} & 0 \end{bmatrix}. \tag{4b}$$

For notational convenience, we denote

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} \in \mathbb{R}^6$$

as the concatenation of the translational and angular velocity vectors, and rewrite (3) equivalently as

$$\dot{\bar{\eta}} = J(\bar{\eta})V - S(\bar{\eta})w. \tag{5}$$

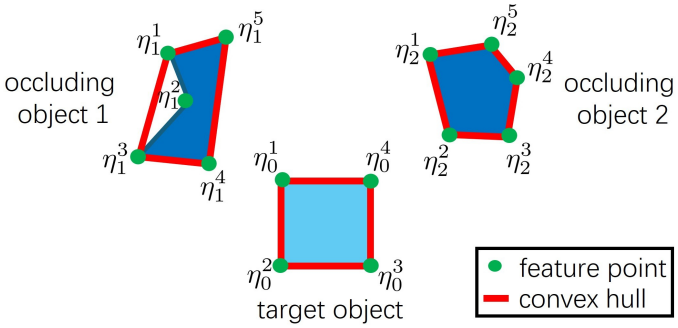


Figure 2: Illustration of the vertex-based representation of target and occluding object images. The target object (cyan) and two occluding objects (blue) are over-approximated by the convex hulls (red edges) of their corresponding feature points (green dots): $\mathbb{T} \subseteq \text{conv}(\eta_0^1, \cdots, \eta_0^4)$, $\mathbb{O}_1 \subseteq \text{conv}(\eta_1^1, \cdots, \eta_1^5)$, and $\mathbb{O}_2 \subseteq \text{conv}(\eta_2^1, \cdots, \eta_2^5)$.

B. Vertex-Based Representation of Object Images

In computer vision, an image feature refers to any structural attribute extracted from an image, and a good feature point is one that can be located unambiguously across different views [10]. In this work, the images of the target and occluding objects are assumed to be over-approximated by the convex hulls of their respective feature points.

Let $\eta_0^i \in \mathbb{R}^2$ denote the image coordinates of the i-th feature point on the target object, where $i \in [N_0]$ with $N_0 \in \mathbb{Z}_{>0}$; let η_j^i denote the image coordinates of the i-th feature point on the j-th occluding object, where $i \in [N_j], j \in [M]$ with $N_j, M \in \mathbb{Z}_{>0}$. Denote by $\mathbb{T}$ the image of the target object and by $\mathbb{O}_j$ the image of the j-th occluding object. Then,

$$\mathbb{T} \subseteq \text{conv}(\eta_0^1, \cdots, \eta_0^{N_0}) \triangleq \bar{\mathbb{T}}, \tag{6a}$$

$$\mathbb{O}_j \subseteq \text{conv}(\eta_j^1, \cdots, \eta_j^{N_j}) \triangleq \bar{\mathbb{O}}_j, \ j \in [M], \tag{6b}$$

where $\text{conv}(\cdot)$ denotes the convex hull operator (see Figure 2). Note that N_0 and N_j may vary depending on the specific image processing algorithm used to extract the feature point coordinates. Nevertheless, the control framework presented in Sections III and IV remains valid as long as (6) holds.

In contrast to existing approaches that rely on the half-space representation (H-rep) of convex polytopes [17], [21], we adopt the *vertex representation* (V-rep) [22] in our control framework. This choice offers several advantages: it aligns directly with image data, enhances numerical robustness, and enables a differentiable formulation. More detailed discussions are provided in Section III.

C. Control Barrier Functions

Consider a control affine system $\dot{x} = f(x) + g(x)u$, where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are known and locally Lipchitz continuous functions. Define a safe set

$$\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \tag{7}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function that satisfies $\frac{\partial h}{\partial x}(x) \neq 0$ for all $h(x) = 0$. The function h is called a CBF of relative degree 1 if $\sup_{u\in\mathbb{R}^m} [L_f h + L_g h u + \gamma h] \geq 0, \ \forall x \in \mathbb{R}^n$, where $\gamma > 0$ is a given positive constant, and $L_f h = \frac{\partial h}{\partial x} f$ and $L_g h = \frac{\partial h}{\partial x} g$ are Lie derivatives [23]. When the CBF condition $L_f h + L_g h u + \gamma h \geq 0$ is incorporated into a Quadratic Program (QP), the resulting CBF-QP-based controller can formally ensure the safety (in the sense of forward invariance of $\mathcal{C}$) of the closed-loop system.

D. Problem Statement

Consider the robotic manipulator described as

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \tag{8}$$

where $q \in \mathbb{R}^n$ is the joint angle, $\dot{q} \in \mathbb{R}^n$ is the angular velocity, $\tau \in \mathbb{R}^n$ is the control input, $D : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the inertia matrix, $C : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the Coriolis/centripetal matrix, and $G : \mathbb{R}^n \to \mathbb{R}^n$ is the gravity term [24].

The problem addressed in this article is formally stated as follows:

Problem 1: Consider a robotic manipulator equipped with a camera at its end-effector, where the robot dynamics are given by (8) and the visual dynamics by (5). The goal is to design a control law for the robot that achieves the following two tasks simultaneously:

1) *Target feature points regulation*: The image coordinates of all feature points on the target object, $\eta_0^i \in \mathbb{R}^2$, are regulated to their desired values, $\eta_{0,d}^i$, i.e.,

$$\eta_0^i(t) \to \eta_{0,d}^i(t) \text{ as } t \to \infty, \ \forall i \in [N_0]. \tag{9}$$

2) *Occlusion avoidance*: Visual occlusion with any of the M occluding objects is avoided during the execution, i.e.,

$$\mathbb{O}(t) \cap \mathbb{T}_j(t) = \emptyset, \ \forall t \geq 0, \forall j \in [M]. \tag{10}$$

We propose a hierarchical, optimization-based, and occlusion-free visual servoing framework to solve Problem 1. As illustrated in Figure 1, the control scheme comprises two main components: a high-level MPC (Section III) and a low-level CBF-based controller (Section IV). The high-level MPC ensures target regulation and provides a desired input to the CBF-based controller, while the low-level controller strictly enforces occlusion-avoidance constraints. The reference velocity generated by the CBF-based controller is then tracked by a standard Cartesian velocity controller to yield the final control input τ for the robot.

III. High-level MPC via Duality-Based Optimization

In this section, we present an MPC as the high-level control scheme of the hierarchical framework, which generates the desired Cartesian velocity in discrete time, enabling the robot to regulate target feature points (Task 1 of Problem 1) and avoid visual occlusion. By leveraging a vertex-based representation of object images and duality theory in optimization, the nonsmooth distance constraints are reformulated as smooth functions that can be seamlessly incorporated into the MPC.

The discrete-time MPC model of the visual dynamics is obtained by applying explicit Euler discretization to (5). For the i-th feature point on the j-th object, the dynamic model is

$$\bar{\eta}_j^i(k+1) = \bar{\eta}_j^i(k) + TJ(\bar{\eta}_j^i(k))V(k) - TS(\bar{\eta}_j^i(k))w_j^i(k), \quad (11)$$

where k is the discrete time step, T is the sampling period, and $\bar{\eta}_j^i$ and w_j^i are the augmented state and velocity of the i-th feature point on the j-th object, respectively. Recall that $j = 0$ in η_j^i corresponds to the target object, while $j \in [M]$ indexes the occluding objects. The feature point velocity w_j^i must be estimated over the prediction horizon (see Remark 1).

We use the distance between sets to formulate the occlusion-avoidance constraints $\mathbb{O}(t) \cap \mathbb{T}_j(t) = \emptyset$ shown in (10). The distance between the target object and the j-th occluding object is defined as

$$\mathrm{d}(\mathbb{T}, \mathbb{O}_j) = \min_{x,y} \{\|x - y\| : x \in \mathbb{T}, y \in \mathbb{O}_j\}. \quad (12)$$

It is obvious that $\mathrm{d}(\mathbb{T}, \mathbb{O}_j) > 0$ implies $\mathbb{O}(t) \cap \mathbb{T}_j(t) = \emptyset$. With the vertex representations of the convex hulls defined in (6), the distance $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$ can be obtained by solving the following optimization problem:

$$\begin{aligned} \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) = \min_{\alpha,\beta,\theta} \ & \|\theta\| && (13a)\\ \text{s.t.} \ & \alpha \geq 0, \beta \geq 0, && (13b)\\ & \mathbf{1}_{N_0}^\top \alpha = 1, \mathbf{1}_{N_j}^\top \beta = 1, && (13c)\\ & \Gamma\alpha - \Lambda_j \beta = \theta, && (13d) \end{aligned}$$

where $\alpha \in \mathbb{R}^{N_0}$, $\beta \in \mathbb{R}^{N_j}$, and $\theta \in \mathbb{R}^2$ are decision variables,

$$\Gamma = [\eta_0^1, \eta_0^2, \cdots, \eta_0^{N_0}], \quad (14)$$

$$\Lambda_j = [\eta_j^1, \eta_j^2, \cdots, \eta_j^{N_j}], \ j \in [M]. \quad (15)$$

Since $\mathrm{d}(\mathbb{T}, \mathbb{O}_j) \geq \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$, the occlusion-avoidance constraints $\mathbb{O}(t) \cap \mathbb{T}_j(t) = \emptyset$ shown in (10) can be relaxed to $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) > 0$. We add a safety margin so that the relaxed occlusion-avoidance constraints are given as

$$\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) \geq d_{\min}, \quad (16)$$

for any $j \in [M]$, where $d_{\min} > 0$ denotes the safety margin. However, condition (16) cannot be directly incorporated in an MPC since $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$ involves solving the optimization problem (13) and its Jacobian and Hessian are hard to obtain. To address this issue, inspired by [25], we formulate a set of differentiable constraints to enforce the occlusion-avoidance constraints (16), as shown in the following theorem.

Theorem 1: The occlusion-avoidance constraint (16) is satisfied if there exist $\lambda_j \in \mathbb{R}^{N_0}$, $\mu_j \in \mathbb{R}^{N_j}$, $\xi_j, \zeta_j \in \mathbb{R}$, and $\nu_j \in \mathbb{R}^2$ such that the following conditions hold true:

$$\begin{aligned} & -\xi_j - \zeta_j \geq d_{\min}, && (17a)\\ & \lambda_j \geq 0, \mu_j \geq 0, && (17b)\\ & \nu_j^\top \nu_j \leq 1, && (17c)\\ & \Gamma^\top \nu_j - \lambda_j + \xi_j \mathbf{1}_{N_0} = 0, && (17d)\\ & -\Lambda_j^\top \nu_j - \mu_j + \zeta_j \mathbf{1}_{N_j} = 0. && (17e) \end{aligned}$$

Proof: The Lagrangian of optimization problem (13) is

$$\begin{aligned} L(\lambda_j, \mu_j, \xi_j, \nu_j, \zeta_j, \alpha, \beta, \theta) &= \|\theta\| - \lambda_j^\top \alpha - \mu_j^\top \beta \\ &+ \xi_j(\mathbf{1}_{N_0}^\top \alpha - 1) + \zeta_j(\mathbf{1}_{N_j}^\top \beta - 1) + \nu_j^\top(\Gamma\alpha - \Lambda_j\beta - \theta). \end{aligned}$$

Therefore, the Lagrange dual function is

$$\begin{aligned} g(\lambda_j, \mu_j, \xi_j, \nu_j, \zeta_j) &= \inf_{\alpha,\beta,\theta} L(\lambda_j, \mu_j, \xi_j, \nu_j, \zeta_j, \alpha, \beta, \theta) \\ &= -\xi_j - \zeta_j + \inf_\theta \left(\|\theta\| - \nu_j^\top \theta\right) \\ &\quad + \inf_\alpha (\nu_j^\top \Gamma - \lambda_j^\top + \xi_j \mathbf{1}_{N_0}^\top)\alpha \\ &\quad + \inf_\beta (-\nu_j^\top \Lambda_j - \mu_j^\top + \zeta_j \mathbf{1}_{N_j}^\top)\beta. \end{aligned}$$

Note that $\inf_\theta \left(\|\theta\| - \nu_j^\top \theta\right) = -f^*(\nu)$ where $f(\theta) = \|\theta\|$ and f^* is the convex conjugate. Since the conjugate of the norm is the indicator function of the unit ball of the dual norm, we have $f^*(\nu) = 0$ if $\|\nu_j\| \leq 1$ and $+\infty$ if $\|\nu_j\| > 1$, which implies $\inf_\theta \left(\|\theta\| - \nu_j^\top \theta\right) = 0$ if $\|\nu_j\| \leq 1$ and $-\infty$ if $\|\nu_j\| > 1$. It is clear that $\inf_\alpha (\nu_j^\top \Gamma - \lambda_j^\top + \xi_j \mathbf{1}_{N_0}^\top)\alpha = 0$ if $\nu_j^\top \Gamma - \lambda_j^\top + \xi_j \mathbf{1}_{N_0}^\top = 0$, and $-\infty$ otherwise; similarly, $\inf_\beta (-\nu_j^\top \Lambda_j - \mu_j^\top + \zeta_j \mathbf{1}_{N_j}^\top)\beta = 0$ if $-\nu_j^\top \Lambda_j - \mu_j^\top + \zeta_j \mathbf{1}_{N_j}^\top = 0$, and $-\infty$ otherwise. Therefore, the dual problem of (13) is

$$\begin{aligned} \tilde{d}_j = \max_{\lambda_j, \mu_j, \xi_j, \zeta_j, \nu_j} \ & -\xi_j - \zeta_j && (18)\\ \text{s.t.} \ & \lambda_j \geq 0, \mu_j \geq 0, \\ & \|\nu_j\| \leq 1, \\ & \nu_j^\top \Gamma - \lambda_j^\top + \xi_j \mathbf{1}_{N_0}^\top = 0, \\ & -\nu_j^\top \Lambda_j - \mu_j^\top + \zeta_j \mathbf{1}_{N_j}^\top = 0. \end{aligned}$$

Thus, the conditions in (17) imply $\tilde{d}_j \geq d_{\min}$. By the weak duality, $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) \geq \tilde{d}_j \geq d_{\min}$. This completes the proof. □

In contrast to existing works [17], [25], which employ the H-rep of polytopes for occlusion constraints, Theorem 1 establishes equivalent conditions using the vertex representation. The V-rep offers several important advantages over the H-rep for handling the image-based occlusion avoidance.

- *Direct compatibility with image data*: Image processing algorithms typically provide only the pixel coordinates of feature points, which usually correspond to boundary points. Using V-rep eliminates the need of computing half-space conversion, which can be computationally expensive.
- *Numerical robustness*: H-rep can suffer from numerical issues when some feature points lie inside the polytope, potentially leading to redundant or ill-conditioned constraints. In contrast, V-rep is less sensitive to such degeneracies.
- *Differentiability and analytical convenience*: If the H-rep of the target object's image is expressed as $\{\eta \in \mathbb{R}^2 \mid A(\eta_0^1, \ldots, \eta_0^{N_0})\eta \leq b(\eta_0^1, \ldots, \eta_0^{N_0})\}$, then the gradients of A and b are difficult or impossible to compute, posing a significant challenge for CBF-based control design. V-rep avoids this issue by maintaining a smooth and direct dependence on the coordinates of feature points.

By incorporating the dual-form safety constraints (17) in discrete time, we formulate the MPC shown in (19) to regulate the target feature points to their desired values. In (19a), the positive definite matrices Q, R, Q_N penalize the stage states, control inputs, and terminal states, respectively;

$$\min_{\substack{\{\bar{\eta}_j^i(n|k),V(n|k),\\ \xi_{jn},\zeta_{jn},\nu_{jn},\lambda_{jn},\mu_{jn}\},\\ j\in[0,M],i\in[N_i],n\in[0,N]}} \sum_{n=0}^{N-1}\sum_{i=1}^{N_0}(\|[\bar{\eta}_0^i(n|k)]_{1:2}-\eta_{0,d}^i\|_Q^2+\|V(n|k)\|_R^2)+\sum_{i=1}^{N_0}\|[\bar{\eta}_0^i(N|k)]_{1:2}-\eta_{0,d}^i\|_{Q_N}^2 \tag{19a}$$

$$\text{s.t.} \quad \bar{\eta}_j^i(0|k)=\bar{\eta}_j^i(k),\ j\in[0,M],i\in[N_i], \tag{19b}$$

$$\bar{\eta}_j^i(n+1|k)=\bar{\eta}_j^i(n|k)+TJ(\bar{\eta}_j^i(n|k))V(n|k) -TS(\bar{\eta}_j^i(n|k))\hat{w}_j^i(k),\ j\in[0,M],i\in[N_i],n\in[0,N], \tag{19c}$$

$$-\xi_{jn}-\zeta_{jn}\geq d_{\min},\ j\in[M],n\in[N], \tag{19d}$$

$$\lambda_{jn}\geq 0,\mu_{jn}\geq 0,\|\nu_{jn}\|\leq 1,\ j\in[M],n\in[N], \tag{19e}$$

$$\Gamma_n^\top\nu_{jn}-\lambda_{jn}+\xi_{jn}\mathbf{1}_{N_0}=0,\ j\in[M],n\in[N], \tag{19f}$$

$$-\Lambda_{jn}^\top\nu_{jn}-\mu_{jn}+\zeta_{jn}\mathbf{1}_{N_j}=0,\ j\in[M],n\in[N], \tag{19g}$$

$$V_{\min}\leq V(n|k)\leq V_{\max},\ n\in[0,N], \tag{19h}$$

$$\eta_{\min}\leq\eta_0^i(n|k)\leq\eta_{\max},\ n\in[0,N],i\in[N_0]. \tag{19i}$$

$[\bar{\eta}_0^i(n|k)]_{1:2}$, which is equal to $\eta_0^i(n|k)$, represent the image coordinates of the i-th feature point on the target; and $\eta_{0,d}^i$ is the desired image coordinates of the target. Condition (19b) specifies the initial condition of $\bar{\eta}_j^i$; condition (19c) is the system dynamics constraint from (11); conditions (19d)-(19g) impose the occlusion-avoidance constraints from (17); condition (19h) bounds the control inputs, where $V_{\min},V_{\max}\in\mathbb{R}^6$ denote the minimum and maximum end-effector velocities; and condition (19i) imposes the field-of-view constraints, where $\eta_{\min},\eta_{\max}\in\mathbb{R}^2$ denote the bounds on the pixel coordinates of target feature points. The term Γ_n in (19f) and Λ_{jn} in (19g) are defined according to (14) and (15): $\Gamma_n=[\eta_0^1(n|k),\eta_0^2(n|k),\cdots,\eta_0^{N_0}(n|k)]$, $\Lambda_{jn}=[\eta_j^1(n|k),\eta_j^2(n|k),\cdots,\eta_j^{N_j}(n|k)]$, and the term $\hat{w}_j^i$ in (19c) denotes the estimated velocity of the i-th feature point of object j. In the MPC problem (19), the total number of constraints is $6+5N_0+3N_0N+(4+N_0)MN+(3+4N)\sum_{i=1}^M N_i$, and the number of decision variables is $N\left(6+(4+N_0)M+\sum_{i=1}^M N_i\right)$; therefore, the problem size of (19) scales polynomially with respect to the number of vertices, occluding objects, and the prediction horizon.

The MPC problem (19) is solved in a receding horizon manner at each discrete time step. At time k, the end-effector velocity $V(0|k)$ obtained from (19) is provided to the low-level CBF as the nominal input V_{mpc}. Although the occlusion constraint (16) is enforced only at discrete time instants in (19), its continuous satisfaction is ensured by the low-level CBF condition, which will be described in Section IV.

Remark 1: In practice, we assume that the estimated velocity $\hat{w}_j^i$ in (19c) remains constant over the prediction horizon and determine its value from the measurement of $\bar{\eta}_j^i$. Below, we present two methods for estimating w using the state measurement $\bar{\eta}$ based on the visual dynamics (5).

Extended Kalman Filter (EKF). Note that the visual dynamics (5) can be discretized and reformulated as

$$\begin{bmatrix}\bar{\eta}(k+1)\\ w(k+1)\end{bmatrix}=\begin{bmatrix}\bar{\eta}(k)+TJ(\bar{\eta}(k))V(k)-TS(\bar{\eta}(k))w(k)\\ w(k)+T\dot{w}(k)\end{bmatrix},$$
$$y(k)=\bar{\eta}(k),$$

where T is the sampling time, $w(k)$ is the state variable to be estimated, $y(k)$ is the output measurement, and $\dot{w}(k)$ is treated as process noise. An EKF can then be deployed to simultaneously estimate $\bar{\eta}$ and w.

Disturbance Observer (DOB). Suppose that the acceleration of the feature point satisfies $\|\dot{w}\|\leq W$, where $W>0$ is a constant, and that $z(t)\neq 0$ for any $t\geq 0$. These assumptions are mild and readily satisfied in practice. Under these conditions, the DOB from [26] can be employed to obtain $\hat{w}$ as the estimate of w,

$$\hat{w}=m+\alpha p,$$
$$\dot{m}=-\alpha L_d\left(J(\bar{\eta})V-S(\bar{\eta})\hat{w}\right),$$

where $\alpha>\frac{1}{2}$ is a constant, and

$$p=[-\frac{u_xz}{f},-\frac{u_yz}{f},-z]^\top, L_d=\begin{bmatrix}-\frac{z}{f}&0&-\frac{u_x}{f}\\ 0&-\frac{z}{f}&-\frac{u_y}{f}\\ 0&0&-1\end{bmatrix}.$$

We define the estimation error as $\tilde{w}=\hat{w}-w$ and consider the candidate Lyapunov function $\mathrm{V}=\frac{1}{2}\tilde{w}^\top\tilde{w}$. It follows that $\dot{\mathrm{V}}=-\alpha\tilde{w}^\top\tilde{w}-\tilde{w}^\top\dot{w}\leq-\tilde{w}^\top\dot{w}+\|\tilde{w}\|W\leq-\left(\alpha-\frac{1}{2}\right)\|\tilde{w}\|^2+\frac{W^2}{2}=-(2\alpha-1)\mathrm{V}+\frac{W^2}{2}$, which implies that $\tilde{w}$ is uniformly ultimately bounded [27].

The two methods above each have their advantages and limitations. With a sufficiently high measurement update frequency, the DOB can estimate w accurately, and its estimation error bound is known and can be incorporated into the control design. However, in our experiments, the camera update rate is limited to 30 Hz, resulting in insufficient estimation accuracy. By contrast, the EKF performs well empirically, but its estimation error cannot be explicitly quantified.

Remark 2: The optimization problem (19) may become infeasible, particularly when the distance between the target and occluding objects approaches $d_{\min}$. To address this issue, we introduce slack variables $\delta_j(n|k)$, which satisfies $0\leq\delta_j(n|k)\leq d_{\min}$, into constraint (19d), reformulating it as

$$-\xi_j(n|k)-\zeta_j(n|k)\geq d_{\min}-\delta_j(n|k). \tag{20}$$

A quadratic penalty term $p\delta_j(n|k)^2$, with $p > 0$, is then added to the objective function (19a). While this relaxation may allow occlusion when MPC is applied alone, occlusion avoidance is still guaranteed under the proposed hierarchical control framework, as the low-level CBFs strictly enforce the occlusion-avoidance constraints.

The variable z in (4) appears in the denominators, which introduces strong nonlinearities into the model. In practice, however, the depth typically varies slowly relative to the MPC update rate, resulting in only small variations over a single MPC horizon. Therefore, treating z as fixed within each MPC iteration provides a reasonable local approximation. Specifically, upon receiving state feedback and initiating the MPC computation, the control sequence V from the previous iteration is used to forward propagate a sequence of z via the visual dynamics (5). Then, in (19), the values $z(n|k)$, $n \in [N]$, are set to this sequence and treated as known constants rather than decision variables.

Remark 3: Condition (10) is sufficient but not necessary for avoiding occlusion in practice. The resulting conservatism arises from the simplified convex geometric representation and the imposed strict separation assumptions. In future work, we plan to extend our method to mitigate this conservatism in the occlusion-avoidance condition, while preserving the theoretical rigor and computational tractability of the proposed framework.

IV. Low-Level CBF-QP-based Safe Control

In this section, we present a CBF-QP-based controller as the low-level scheme of the hierarchical framework, which strictly enforces the occlusion-avoidance constraint (16) for all $t \geq 0$ in continuous time, thereby achieving Task 2 of Problem 1.

We consider the image coordinate dynamics of the i-th feature point on object j from (5),

$$\dot{\eta}_j^i = \bar{J}(\bar{\eta}_j^i)V - \bar{S}(\bar{\eta}_j^i)w_j^i, \tag{21}$$

where $\eta_j^i \in \mathbb{R}^2$ denotes the image coordinates defined in (1), and $\bar{J}(\bar{\eta}_i^j)$ and $\bar{S}(\bar{\eta}_i^j)$ consist of the first two rows of $J(\bar{\eta}_i^j)$ and $S(\bar{\eta}_i^j)$, respectively, from (4). Compared with the model (5) used for the MPC design, the model (21) excludes the dynamics of z since they are not required in the CBF design.

To compactly express $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$, the distance between $\bar{\mathbb{T}}$ and $\bar{\mathbb{O}}_j$, we define the vector obtained by concatenating all feature points of $\bar{\mathbb{T}}$ and $\bar{\mathbb{O}}_j$,

$$\boldsymbol{\eta}_j = \quad [\eta_0^{1\top}, \cdots, \eta_0^{N_0\top}, \eta_j^{1\top}, \cdots, \eta_j^{N_j\top}]^\top \in \mathbb{R}^{2(N_0+N_j)}.$$

From (21), the dynamics of $\boldsymbol{\eta}_j$ can be expressed as

$$\dot{\boldsymbol{\eta}}_j = J_j(\bar{\boldsymbol{\eta}}_j)V - S_j(\bar{\boldsymbol{\eta}}_j)\boldsymbol{w}_j, \tag{22}$$

where $\bar{\boldsymbol{\eta}}_j = [\bar{\eta}_0^{1\top}, \cdots, \bar{\eta}_0^{N_0\top}, \bar{\eta}_j^{1\top}, \cdots, \bar{\eta}_j^{N_j\top}]^\top \in \mathbb{R}^{3(N_0+N_j)}$, $J_j = [\bar{J}(\bar{\eta}_0^1)^\top, \cdots, \bar{J}(\bar{\eta}_0^{N_0})^\top, \bar{J}(\bar{\eta}_j^1)^\top, \cdots, \bar{J}(\bar{\eta}_j^{N_j})^\top]^\top$, $S_j(\bar{\boldsymbol{\eta}}_j) = \mathrm{blkdiag}(\bar{S}(\bar{\boldsymbol{\eta}}_0^1), \cdots, \bar{S}(\bar{\boldsymbol{\eta}}_0^{N_0}), \bar{S}(\bar{\boldsymbol{\eta}}_j^1), \cdots, \bar{S}(\bar{\boldsymbol{\eta}}_j^{N_j}))$, and $\boldsymbol{w}_j = [w_0^{1\top}, \cdots, w_0^{N_0\top}, w_j^{1\top}, \cdots, w_j^{N_j\top}]^\top$, $j \in [M]$.

We aim to design CBFs that ensure the occlusion-avoidance constraints (16) for all $t \geq 0$ and $j \in [M]$. However, $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$ is computed by solving the optimization problem (13), whose objective function is not strictly convex. This leads to a non-unique minimizer and a non-Lipschitz continuous distance function, which poses challenges in computing the gradient of $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)$. To address this issue, we define the following function $\tilde{h}_j(\boldsymbol{\eta}_j)$ for any $j \in [M]$:

$$\begin{aligned} \tilde{h}_j(\boldsymbol{\eta}_j) = \min_{\alpha,\beta,\theta} \ & \theta^\top\theta + \delta_1\alpha^\top\alpha + \delta_2\beta^\top\beta \\ \text{s.t.} \ & \text{(13b)} - \text{(13d)}, \end{aligned} \tag{23}$$

where $\delta_1, \delta_2 > 0$ are positive constants representing the penalty weights. It is clear that $\tilde{h}_j \geq \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)^2$. Furthermore, although $\tilde{h}_j$ can be shown to be Lipschitz continuous with respect to $\boldsymbol{\eta}_j$, it may be nondifferentiable at points where the strict complementarity condition fails for the optimization problem (23) [28]. To address this issue, we employ nonsmooth CBF techniques [29], [30].

For a set A, define $\bar{\mathcal{P}}(A) = \mathcal{P}(A)\backslash A$, where $\mathcal{P}(A)$ denotes the power set of A. Given two sets $\mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0]), \mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])$, and any $j \in [M]$, we define a function $F_j(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)$ as

$$\begin{aligned} F_j(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) = \min_{\alpha,\beta,\theta} \ & \theta^\top\theta + \delta_1\alpha^\top\alpha + \delta_2\beta^\top\beta && \text{(24a)} \\ \text{s.t.} \ & \alpha \geq 0, \ \beta \geq 0, && \text{(24b)} \\ & \alpha_k = 0, \ \forall k \in \mathcal{I}_\alpha, && \text{(24c)} \\ & \beta_l = 0, \ \forall l \in \mathcal{I}_\beta, && \text{(24d)} \\ & \mathbf{1}_{N_0}^\top\alpha = 1, \mathbf{1}_{N_j}^\top\beta = 1, && \text{(24e)} \\ & \Gamma\alpha - \Lambda_j\beta = \theta. && \text{(24f)} \end{aligned}$$

The following result characterizes the differentiability of F_j and its relation to $\tilde{h}_j$.

Theorem 2: Consider the functions $\tilde{h}_j$ defined in (23) and F_j defined in (24). The following statements hold true:

1) The function F_j is well-defined and differentiable for any $\boldsymbol{\eta}_j$, $\mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0])$, and $\mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])$.
2) For any $\boldsymbol{\eta}_j$, there exists $\mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0])$ and $\mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])$ such that $F_j(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) = \tilde{h}_j(\boldsymbol{\eta}_j)$.

Proof: Given $\mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0])$ and $\mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])$, we define a matrix

$$M(\mathcal{I}_\alpha, \mathcal{I}_\beta) = \begin{bmatrix} -[\mathrm{I}_{N_0}]_{\mathcal{I}_\alpha} & 0 & 0 \\ 0 & -[\mathrm{I}_{N_j}]_{\mathcal{I}_\beta} & 0 \\ \mathbf{1}_{N_0}^\top & 0 & 0 \\ 0 & \mathbf{1}_{N_j}^\top & 0 \\ \Gamma & -\Lambda_j & -\mathrm{I}_2 \end{bmatrix}, \tag{25}$$

where $[\mathrm{I}_{N_0}]_{\mathcal{I}_\alpha}$ and $[\mathrm{I}_{N_j}]_{\mathcal{I}_\beta}$ denote the matrix formed by selecting the rows of I_{N_0} and I_{N_j} indexed by the sets $\mathcal{I}_\alpha$ and $\mathcal{I}_\beta$, respectively. We will first show that the linear independence constraint qualification (LICQ) is satisfied for the optimization problems (23) and (24). Suppose the solution to (23) is $(\alpha^*, \beta^*, \theta^*)$ and the corresponding active sets are $\mathcal{I}_\alpha^* = \{i \in [N_0] : \alpha_i^* = 0\}$ and $\mathcal{I}_\beta^* = \{i \in [N_j] : \beta_i^* = 0\}$. Note that LICQ holds if the matrix $M(\mathcal{I}_\alpha^*, \mathcal{I}_\beta^*)$ has full row rank. It is straightforward to verify that $\alpha_i^* = 0$ cannot hold for all $i \in [N_0]$ (i.e., $\mathcal{I}_\alpha^* \neq [N_0]$); otherwise the equality constraints $\mathbf{1}_{N_0}^\top\alpha = 1$ cannot be satisfied. A similar argument applies to the constraints $\beta \geq 0$. Hence, the matrix defined in

(25) has full row rank, indicating that LICQ is satisfied. It can be shown that LICQ holds for (24) by similar reasoning.

1) Since the QP (24) is strictly convex, the value of F_j is uniquely determined for a given $\boldsymbol{\eta}_j$, implying that F_j is well defined. Moreover, because the LICQ condition holds for (24), the optimal multiplier is also unique [31, Section 12.2], and the KKT conditions can be written as

$$\underbrace{\begin{bmatrix} Qz^* + A^\top(\boldsymbol{\eta}_j)\lambda_A^* + G^\top\lambda_G^* \\ Gz^* \\ A(\boldsymbol{\eta}_j)z^* - b \end{bmatrix}}_{\triangleq R(z^*,\lambda_G^*,\lambda_A^*,\boldsymbol{\eta}_j)} = 0. \tag{26}$$

where $z^* = [\theta^{*\top}, \alpha^{*\top}, \beta^{*\top}]^\top$, λ_A^*, λ_G^* denote the multipliers, $Q = \text{blkdiag}(2\mathrm{I}_2, 2\delta_1\mathrm{I}_{N_0}, 2\delta_2\mathrm{I}_{N_j})$, $b = [0, 0, 1, 1]^\top$, and

$$G = \begin{bmatrix} -[\mathrm{I}_{N_0}]_{\mathcal{I}_\alpha} & 0_{N_0\times N_j} & 0_{N_0\times 2} \\ 0_{N_j\times N_0} & -[\mathrm{I}_{N_j}]_{\mathcal{I}_\beta} & 0_{N_j\times 2} \end{bmatrix}, \tag{27a}$$

$$A(\boldsymbol{\eta}_j) = \begin{bmatrix} \Gamma & -\Lambda_j & -\mathrm{I}_2 \\ \mathbf{1}_{N_0}^\top & 0_{N_j\times 1} & 0_{1\times 2} \\ 0_{N_0\times 1} & \mathbf{1}_{N_j}^\top & 0_{1\times 2} \end{bmatrix}. \tag{27b}$$

Let $x^* = [z^{*\top}, \lambda_A^{*\top}, \lambda_G^{*\top}]^\top$. Then,

$$\frac{\partial R}{\partial x^*} = \begin{bmatrix} Q & A^\top & G^\top \\ G & 0 & 0 \\ A & 0 & 0 \end{bmatrix}. \tag{28}$$

Since Q is positive definite and $[G^\top, A^\top]^\top$ has full row rank following the argument after (25), it is easy to show that $\frac{\partial R}{\partial x^*}$ is nonsingular for any $\boldsymbol{\eta}_j \in \mathbb{R}^{2(N_0+N_j)}$ [31, Lemma 16.1]. According to the implicit function theorem, one can see that F_j, which corresponds to the solution to (24), is differentiable.

2) The second statement follows from the fact that $F_j(\boldsymbol{\eta}_j; \mathcal{I}_\alpha^*, \mathcal{I}_\beta^*) = \tilde{h}_j(\boldsymbol{\eta}_j)$, where $\mathcal{I}_\alpha^*$ and $\mathcal{I}_\beta^*$ are active sets of the problem (23). □

From Theorem 2, the gradient of $F_j(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)$ with respect to $\boldsymbol{\eta}_j$ can be computed, which is required for the implementation of the CBF controller. Specifically, denote the i-th entry of $\boldsymbol{\eta}_j$ by $\boldsymbol{\eta}_{ji}$, $i \in [2N_0 + 2N_j]$, and take the partial derivative with respect to $\boldsymbol{\eta}_{ji}$ on both sides of (26),

$$\begin{bmatrix} Q & A^\top(\boldsymbol{\eta}_j) & G^\top \\ G & 0 & 0 \\ A(\boldsymbol{\eta}_j) & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial z^*}{\partial \boldsymbol{\eta}_{ji}} \\ \frac{\partial \lambda_A^*}{\partial \boldsymbol{\eta}_{ji}} \\ \frac{\partial \lambda_G^*}{\partial \boldsymbol{\eta}_{ji}} \end{bmatrix} = \begin{bmatrix} -\frac{\partial A^\top(\boldsymbol{\eta}_j)}{\partial \boldsymbol{\eta}_{ji}}\lambda_A^* \\ 0 \\ -\frac{\partial A^\top(\boldsymbol{\eta}_j)}{\partial \boldsymbol{\eta}_{ji}} z^* \end{bmatrix}. \tag{29}$$

Since Q is positive definite and $[G^\top, A^\top]^\top$ has full row rank, the linear system (29) admits a unique solution. Collecting the solutions for all entries of $\boldsymbol{\eta}_j$, we define $\frac{\partial z^*}{\partial \boldsymbol{\eta}_j} = \begin{bmatrix} \frac{\partial z^*}{\partial \boldsymbol{\eta}_{j1}} & \frac{\partial z^*}{\partial \boldsymbol{\eta}_{j2}} & \cdots & \frac{\partial z^*}{\partial \boldsymbol{\eta}_{j(2N_0+2N_j)}} \end{bmatrix}$. Since $F = \frac{1}{2} z^{*\top} Q z^*$, the gradient $\frac{\partial F}{\partial \boldsymbol{\eta}_j}$ can be computed as

$$\frac{\partial F}{\partial \boldsymbol{\eta}_j} = z^{*\top} Q \frac{\partial z^*}{\partial \boldsymbol{\eta}_j}. \tag{30}$$

From Theorem 2, we can also see that $\tilde{h}_j$ is piecewise differentiable [29, Definition II.9]. By [29, Proposition III.3], for any $\boldsymbol{\eta}_j \in \mathbb{R}^{2(N_0+N_j)}$, there exists a locally encapsulating index set $I_{h_j}(\boldsymbol{\eta}_j) \subseteq \{(\mathcal{I}_\alpha, \mathcal{I}_\beta) : \mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0]), \mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])\}$ such that the generalized gradient of $\tilde{h}_j$ with respect to $\boldsymbol{\eta}_j$ satisfies

$$\partial \tilde{h}_j \subset \text{conv}\left\{ \frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j} : (\mathcal{I}_\alpha, \mathcal{I}_\beta) \in I_{h_j}(\boldsymbol{\eta}_j) \right\}. \tag{31}$$

A straightforward way to compute $I_{h_j}(\boldsymbol{\eta}_j)$ is to check the KKT conditions (26) for all $\mathcal{I}_\alpha \in \bar{\mathcal{P}}([N_0])$ and $\mathcal{I}_\beta \in \bar{\mathcal{P}}([N_j])$.

Based on the results above, we define the following candidate CBF for any $j \in [M]$,

$$h_j(\boldsymbol{\eta}_j) = \tilde{h}_j(\boldsymbol{\eta}_j) - (d_{\min}^2 + \delta_1^2 + \delta_2^2), \tag{32}$$

where $\delta_1, \delta_2 > 0$ are small positive constants. Note that this CBF is implicitly constructed from the solution $\tilde{h}_j$ of optimization (23), which represents a relaxed minimum distance between polytopes [22], [30].

The following result ensures that the occlusion-avoidance constraints (16) are strictly enforced by the CBF-based controller for all $t \geq 0$.

Theorem 3: Consider the system in (22) and the CBF defined in (32). Suppose the feature point velocities are bounded, i.e., $\|w_j^i\| \leq W$, for some constant $W > 0$, $\forall j \in [0, M]$ and $\forall i \in [N_j]$. If $h_j(\boldsymbol{\eta}_j(0)) > 0, \forall j \in [0, M]$, then any Lipschitz controller $V(\boldsymbol{\eta}_j) \in K_{BF}(\boldsymbol{\eta}_j) \triangleq \{V \in \mathbb{R}^6 : \psi_{0,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) + \psi_{1,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)V \geq 0, \forall (\mathcal{I}_\alpha, \mathcal{I}_\beta) \in I_{h_j}(\boldsymbol{\eta}_j)\}$ ensures that the occlusion-avoidance constraints (16) are strictly enforced for all $t \geq 0$, where $\gamma > 0$ is a constant and

$$\psi_{0,j} = \gamma h_j - W\sqrt{N_0 + N_j}\left\| \frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j} S_j(\bar{\boldsymbol{\eta}}_j) \right\|, \tag{33a}$$

$$\psi_{1,j} = \frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j} J_j(\bar{\boldsymbol{\eta}}_j). \tag{33b}$$

Proof: First, we prove that

$$h_j \geq 0 \implies \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) \geq d_{\min}. \tag{34}$$

Let $(\alpha^*, \beta^*, \theta^*)$ and $(\alpha_h^*, \beta_h^*, \theta_h^*)$ be the solutions to the optimization problems (13) and (23), respectively. Clearly, $\|\theta^*\|^2 = \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)^2$, $\tilde{h}_j = \|\theta_h^*\|^2 + \delta_1\|\alpha_h^*\|^2 + \delta_2\|\beta_h^*\|^2$, and $\|\theta^*\|^2 + \delta_1\|\alpha^*\|^2 + \delta_2\|\beta^*\|^2 \geq \|\theta_h^*\|^2 + \delta_1\|\alpha_h^*\|^2 + \delta_2\|\beta_h^*\|^2$.

By the definition of h_j, $h_j \geq 0 \implies \|\theta_h^*\|^2 + \delta_1\|\alpha_h^*\|^2 + \delta_2\|\beta_h^*\|^2 \geq d_{\min}^2 + \delta_1^2 + \delta_2^2$. Therefore,

$$h_j \geq 0 \Rightarrow \|\theta^*\|^2 + \delta_1\|\alpha^*\|^2 + \delta_2\|\beta^*\|^2 \geq d_{\min}^2 + \delta_1^2 + \delta_2^2. \tag{35}$$

Since $\alpha^* \geq 0$ and $\mathbf{1}_{N_0}^\top \alpha^* = 1$, it follows that $0 \leq \alpha_i^* \leq 1$ for $i \in [N_0]$, which implies $0 \leq \alpha_i^{*2} \leq \alpha_i^* \leq 1$, and thus $\|\alpha^*\| \leq 1$. Following the similar argument, $\|\beta^*\| \leq 1$. Therefore, from (35) we have

$$h_j \geq 0 \implies \|\theta^*\|^2 \geq d_{\min}^2,$$

which implies that (34) holds since $\|\theta^*\|^2 = \mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j)^2$.

Next, one can easily see that selecting $V \in K_{BF}$ yields

$$\begin{aligned} &\frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j}(J_j V - S_j \boldsymbol{w}_j) \\ &\geq \frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j} J_j V - \left\| \frac{\partial F(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta)}{\partial \boldsymbol{\eta}_j} S_j \right\| \|\boldsymbol{w}_j\| \\ &\geq -\gamma h_j + \psi_{0,j} + \psi_{1,j} V \geq -\gamma h_j \end{aligned}$$

for any $(\mathcal{I}_\alpha, \mathcal{I}_\beta) \in I_{h_j}(\boldsymbol{\eta}_j)$. Then, from [29, Theorem III.5],

$$\min_{v \in \partial h_j} v^\top (J_j V - S_j \boldsymbol{w}_j) \geq -\gamma h_j, \tag{36}$$

where ∂h_j denotes the generalized gradient of h_j with respect to $\boldsymbol{\eta}_j$. According to [29, Theorem II.7], one can conclude that (36) implies $h_j \geq 0$, which completes the proof. □

By Theorem 3, the filtered end-effector velocity V_{cbf} can be obtained by solving the following CBF-QP:

$$\begin{aligned} V_{\mathrm{cbf}} = \arg\min_{V} \ & \|V - V_{\mathrm{mpc}}\|^2 \\ \text{s.t.} \quad & \psi_{0,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) + \psi_{1,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) V \geq 0, \\ & \forall j \in [M],\ \forall (\mathcal{I}_\alpha, \mathcal{I}_\beta) \in I_{h_j}(\boldsymbol{\eta}_j), \end{aligned} \tag{37}$$

where V_{mpc} is obtained from solving the optimization problem (19) and $\psi_{0,j}, \psi_{1,j}$ are defined in (33).

Finally, the control input τ in (8) can be designed using standard methods to enable the robot's end-effector velocity to track the safe reference velocity V_{cbf}. For example, defining the velocity tracking error defined as $e = V - V_{\mathrm{cbf}}$ with dynamics $\dot{e} = \dot{\mathrm{J}}\dot{q} + \mathrm{J}D^{-1}(\tau - C\dot{q} - G) - \dot{V}_{\mathrm{cbf}}$, where J is the Jacobian, the control input τ can be chosen as $\tau = C\dot{q} + G + D\mathrm{J}^\dagger(-Ke - \dot{\mathrm{J}}\dot{q} + \dot{V}_{\mathrm{cbf}})$ where $\mathrm{J}^\dagger$ denotes the pseudo-inverse of J and $K \in \mathbb{R}^{6\times 6}$ is a positive definite gain matrix.

Remark 4: In practice, the velocity bound W in Theorem 3 is typically chosen empirically. One approach is to assume bounds on the Cartesian velocities of the target and occluding objects and then project these bounds into the image plane to obtain a corresponding bound on W. Alternatively, W can be estimated using the visual dynamics in conjunction with the EKF/DOB (see Remark 1).

Remark 5: If the strict complementarity condition ([31, Definition 12.2]) holds for (23), then $\tilde{h}_j$ is differentiable at the corresponding point [28, Theorem 1], and the encapsulating index set coincides with the active sets of (23). Consequently, the gradient of h_j can be obtained by solving the linear system (29) with $\mathcal{I}_\alpha = \mathcal{I}_\alpha^*$ and $\mathcal{I}_\beta = \mathcal{I}_\beta^*$, where $\mathcal{I}_\alpha^*$ and $\mathcal{I}_\beta^*$ denote the active sets of (23).

Remark 6: In practice, the CBF-QP (37) cannot be solved in continuous time; instead, V is updated in a sampled-data fashion. To compensate for potential safety violations introduced by sampling, a correction term can be added into the CBF condition, modifying the constraint of (37) to

$$\psi_{0,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) + \psi_{1,j}(\boldsymbol{\eta}_j; \mathcal{I}_\alpha, \mathcal{I}_\beta) V \geq \Delta_j$$

where $\Delta_j > 0$ is a constant that compensates for the sampling effect [32], [33], [34], [35].

Remark 7: In the high-level MPC, the velocity w_j^i is estimated, whereas the low-level CBF-QP accounts for its worst-case value. This distinction reflects a trade-off between performance and guaranteed occlusion avoidance. MPC prioritizes performance and operates at discrete time instances; it cannot ensure strict occlusion avoidance, so using velocity estimates - despite potential inaccuracies - is acceptable. When sufficiently accurate, these estimates can improve control performance compared with using the worst-case value. In contrast, the CBF-QP prioritizes strict occlusion avoidance, requiring the worst-case velocity to guarantee safety.

V. Experimental Results

In this section, we present experimental results that validate the proposed hierarchical control framework for occlusion-free visual servoing[1]. All experiments are conducted on a Franka Research 3 manipulator equipped with an Intel RealSense D435i camera mounted on its end-effector, as illustrated in Figures 3 and 4. The camera streams RGB images at 30 Hz with a resolution of 1280×720 pixels. An AprilTag is attached on the end-effector of a UFactory xArm 7 manipulator to serve as the target object. The control framework is implemented on an workstation (Intel Core i5-13500 @ 2.5 GHz, 16 GB RAM) for MPC and CBF computations, and an NVIDIA Jetson AGX Orin (2048 CUDA cores, 64 Tensor Cores) for image processing. The two devices communicate via a direct Ethernet connection using the UDP protocol with 72-byte datagrams.

A. Experimental Setup

1) Object Representation: AprilTag markers from the Tag36h11 family are used to represent both the target and occluding objects. The target is an AprilTag (ID: 0) mounted on the end-effector of the xArm 7 manipulator. Up to three additional AprilTags (IDs: 1–3) are mounted on adjustable support stands and serve as occluding objects.

2) Image Processing: We employ a GPU-accelerated AprilTag detection pipeline implemented in NVIDIA Isaac ROS, fed by an Intel RealSense D435i connected via USB 3.0. RGB images are processed to detect tag corners, rectify tag regions, and decode tag IDs. For each verified tag, a *Perspective-n-Point (PnP)* problem is solved to estimate its 6-DOF pose $\left({}^c p_j, {}^c q_j\right)$, including position and unit-quaternion orientation in the camera frame. From this, the depth of each corner is also obtained. The detection output, $\mathcal{D}_j \coloneqq \left(j,\ \{c_j^i\}_{i=1}^4,\ {}^c p_j,\ {}^c q_j\right)$, provides the ID, image corners, and 3D pose, which are then used by the visual servoing controller.

3) MPC Setup: The high-level MPC is implemented using CasADi [36] with automatic C code generation and solved using IPOPT at 10 Hz. At each time step k, the image coordinates $\bar{\eta}$ of the AprilTag feature points are extracted, and their velocities $\hat{w}_j^i(k)$ are estimated using the EKF described in Remark 1, yielding the filtered states $\bar{\eta}_j^i(k)$. Using these, the point set matrices Γ_n and Λ_{jn} are computed and the optimization problem (19) is solved.

4) CBF-QP Setup: The low-level CBF controller employs OSQP [37] to solve the QP (37), using the CBFs h_j from (32) and the nominal input $V_{\mathrm{mpc}} = V(0|k)$ provided by the high-level MPC. Its solution, V_{cbf}, is sent to the Franka Control Interface to command the robot. The CBF-QP-based controller runs at 30 Hz, synchronized with the camera's RGB image stream.

B. Experimental Results

We consider scenarios with one, two, and three static occluding objects, as well as a scenario with one dynamic

[1]The code is available at https://github.com/wisc-arclab/tcst-safe-vs, and the video of the experimental results is available at https://drive.google.com/file/d/135s5m_QnlaoAmMenyW76TDUD5l6glN2D/view?usp=sharing.

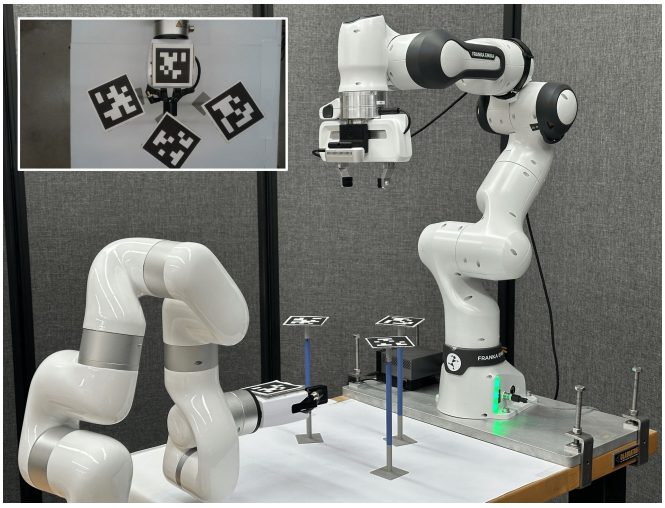


Figure 3: Experimental setup for validating the proposed hierarchical control framework for occlusion-free visual servoing. A Franka Research 3 manipulator (right), equipped with an Intel RealSense D435i camera mounted on its end-effector, tracks a target AprilTag (ID: 0) fixed to the end-effector of a UFactory xArm 7 robot (left). Three occluding AprilTags (IDs: 1, 2, 3) are placed on adjustable-height stands to simulate occlusions. The top-left inset shows the camera view from the RealSense D435i during the experiment.

occluding objects. These scenarios are constructed with progressively increasing occlusion complexity. The proposed hierarchical control framework is evaluated against the high-level MPC (19) alone, i.e., without the low-level CBF-QP controller. For all experiments, the safety margin in (16) is set to $d_{\min} = 20$ (pixels), with δ_1, δ_2 selected as $\delta_1 = \delta_2 = 0.1$. Here, $d_{\min}$ is a tunable parameter chosen to balance experimental clarity, baseline comparability, and robustness to sensing limitations. In (19), the prediction horizon is $N = 7$, and the weight matrices are selected as $Q = \mathrm{I}_8 \in R^{8\times 8}$, $Q_N = 5\mathrm{I}_8 \in R^{8\times 8}$ and $R = 10^6 \times \mathrm{diag}(1.5, 1.5, 1.5, 1, 1, 1) \in \mathbb{R}^{6\times 6}$.

Figure 5 shows snapshots of experiments with static and dynamic occluding objects, while Figure 6 illustrates the evolution of the distances between the target and the obstacles.

*1) **Static Scenarios**:* In the static scenarios, the occluding objects remain stationary along the moving target's path, while the target follows a trajectory toward the Franka manipulator. The velocity bound W in Theorem 3 is set to $W = 0.015$.

Figure 5 (a)-(c) shows camera snapshots for experiments with one, two, and three static occluding objects. AprilTag centers are annotated with their IDs, and tag corners are numbered clockwise from 0 to 3. Occlusion zones are highlighted by yellow squares with a 20-pixel safety margin, meaning the occlusion-avoidance constraint (16) is satisfied whenever the target's feature points remain outside the overlaid yellow squares. The target (AprilTag 0) is enclosed in a green box when persistently visible; the absence of the box indicates loss of recognition due to occlusion. In each scenario, the top row depicts the proposed hierarchical controller (high-level MPC with low-level CBF-QP), while the bottom row shows the high-level MPC alone, where $V(0|k)$ is sent directly as the

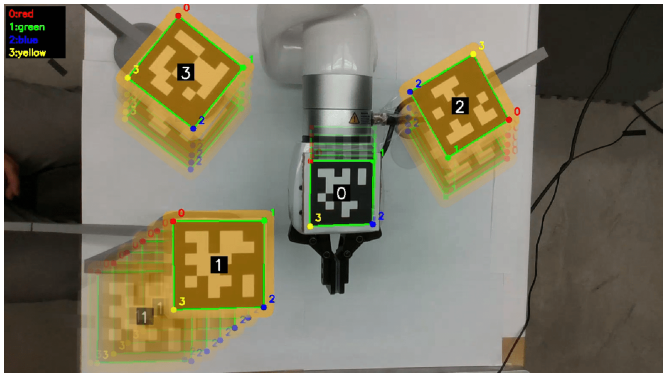


Figure 4: Superimposed camera frames illustrating the dynamic occluding object scenario. The target (AprilTag ID: 0, attached to the xArm gripper) moves along its trajectory while object 1 advances toward it. Objects 2 and 3 remain stationary throughout the experiment.

reference input to the Cartesian velocity tracking controller. Each row presents the initial (left), intermediate (middle), and terminal (right) phases from both the camera and a third-person view.

Figure 6 (a)-(c) shows the distance profiles between the target and the occluding objects for the three static scenarios. The results indicate that the high-level MPC alone fails to consistently enforce the occlusion-avoidance constraint (16): in the one-occluding-object case, the minimum distance drops to $d_{\min} = 12.87$ pixels at $t \approx 4.4$ s; in the two-occluding-object case, the minimum distances are $d_{\min} = 7.43$ pixels at $t \approx 9.3$ s with Object 1 and $d_{\min} = 11.25$ pixels at $t \approx 10$ s with Object 2; and in the three-occluding-object case, the distance between the target and Object 3 reduces to zero at $t \approx 12$ s. By contrast, the hierarchical controller strictly enforces the distances near or above the 20-pixel safety margin in all three scenarios (with only minor single-pixel deviations due to measure noise or hardware disturbances).

*2) **Dynamic Scenario**:* To further demonstrate the real-time reactive capability of the hierarchical control framework, we consider a dynamic scenario in which both the target and an occluding object are moving. Figure 4 shows superimposed camera frames, with three occluding objects (AprilTag ID: 1-3): objects 2 and 3 remain stationary, while object 1 moves toward the target, creating a dynamic occlusion threat. The velocity of objective 1 is estimated using the EKF. The obstacles are manually displaced by dragging the base linkage of the support structure, thereby simulating occluding objects with unpredictable velocities, as often encountered in real-world environments. The velocity bound W in Theorem 3 is set to $W = 0.075$.

Figure 5 (d) shows camera snapshots for the dynamic scenario. As shown in Figure 6 (d), $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_1)$ decreases to 20 at $t \approx 6.6$ s and drops to zero at $t \approx 11.5$ s, resulting in persistent occlusion and target recognition failure. In contrast, the hierarchical controller successfully navigates around the moving obstacle with a considerable margin, strictly enforcing the occlusion-avoidance constraint (16) throughout the entire operation. Objects 2 and 3 exit the camera's field of view at $t \approx 11$ s for both controllers.

Results from the static and dynamic experiments above demonstrate the effectiveness of the hierarchical control framework, where the high-level MPC ensures performance and target regulation, while the low-level CBF-QP controller enforces strict safety by filtering the MPC inputs in real time.

Table I summarizes the computation times for the MPC and CBF-QP controllers across different experimental scenarios, reporting the mean, minimum, maximum, and standard deviation computed from 1000 iterations per scenario. The results confirm that the MPC problem (10 Hz) and the CBF-QP (30 Hz) can each be solved within their respective sampling periods. In particular, the CBF-QP accounts for less than 1% of the total computation time, introducing a nearly negligible computational burden relative to the MPC.

Table I: Computation Times for Different Scenarios

	Controller	Computation Time (ms)			
		Mean	Min.	Max.	Std.
Static Scenarios					
1 occluding object	MPC	23.721	16.450	33.102	3.145
	CBF-QP	0.116	0.085	0.152	0.012
2 occluding objects	MPC	35.311	31.220	40.895	1.988
	CBF-QP	0.205	0.160	0.265	0.021
3 occluding objects	MPC	41.338	29.550	58.112	4.875
	CBF-QP	0.303	0.255	0.380	0.019
Dynamic Scenario					
3 occluding objects	MPC	42.515	35.101	51.440	3.220
	CBF-QP	0.331	0.275	0.410	0.025

VI. Conclusion & Future Work

This article presented a hierarchical, optimization-based visual servoing control scheme for robotic manipulators that guarantees occlusion-free operation by integrating a high-level MPC with a low-level CBF-QP controller. By leveraging the vertex-based representation of object images and a duality-based optimization approach, we reformulated occlusion avoidance into differentiable constraints that can be seamlessly incorporated into both the MPC and CBF design. The resulting framework achieved simultaneous regulation of target feature points and strict continuous-time occlusion avoidance. Hardware experiments on the Franka Research 3 manipulator validated the effectiveness of the proposed approach.

We plan to extend this work along several directions. First, we will mitigate the computational burden introduced by the V-rep formulation by developing culling strategies to eliminate irrelevant occluding objects and leveraging parallel computation techniques to accelerate the MPC. Second, we will investigate the compatibility of multiple CBFs for robotic manipulators to address feasibility issues in the CBF-QP with multiple constraints, exploring both formal characterization of compatibility conditions and the synthesis of a single CBF from multiple occlusion constraints. Finally, we aim to apply the proposed framework to more complex and challenging visual servoing tasks in robotic manipulation.

References

[1] K. Hashimoto, "A review on vision-based control of robot manipulators." *Advanced Robotics*, vol. 17, no. 10, 2003.

[2] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Computers and Electronics in Agriculture*, vol. 127, pp. 311–323, 2016.

[3] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous Robots*, vol. 27, no. 3, pp. 201–219, 2009.

[4] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 4693–4700.

[5] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[6] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813–825, 2002.

[7] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 350–360.

[8] N. Rahimi, S. Talebi, A. Deole, M. Mesbahi, S. Bandyopadhyay, and A. Rahmani, "Robust controller synthesis for vision-based spacecraft guidance and control," in *AIAA SCITECH Forum*, 2022, p. 2213.

[9] C. Hsieh, Y. Li, D. Sun, K. Joshi, S. Misailovic, and S. Mitra, "Verifying controllers with vision-based perception using safe approximate abstractions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4205–4216, 2022.

[10] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 2002.

[11] F. Chaumette and S. Hutchinson, "Visual servo control. I. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[12] ——, "Visual servo control. II. advanced approaches [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[13] P. Geng, M. Luo, T. Li, H. Wang, Y. Qin, and J. Han, "Duality-based optimization of occlusion avoidance for active optical navigation system in robotic orthopedic surgeries," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 15 215–15 226, 2025.

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

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

[16] 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.

[17] S. Wei, B. Dai, R. Khorrambakht, P. Krishnamurthy, and F. Khorrami, "Diffocclusion: Differentiable optimization based control barrier functions for occlusion-free visual servoing," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3235–3242, 2024.

[18] M. Luo, Y. Qin, and J. Han, "Online occlusion-free optimization scheme for active optical navigation system in robotic orthopedic surgeries," *Control Engineering Practice*, vol. 148, p. 105948, 2024.

[19] K. He, R. Newbury, T. Tran, J. Haviland, B. Burgess-Limerick, D. Kulić, P. Corke, and A. Cosgun, "Visibility maximization controller for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8479–8486, 2022.

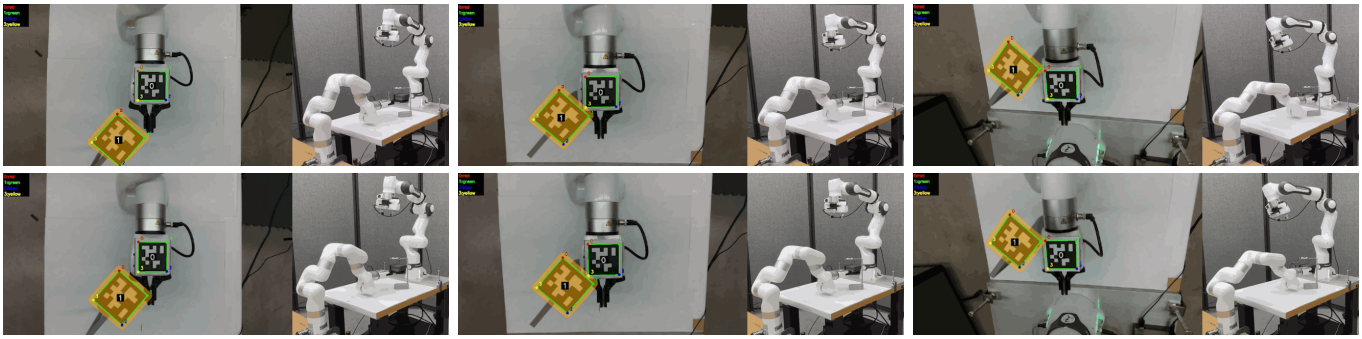
[20] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[21] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

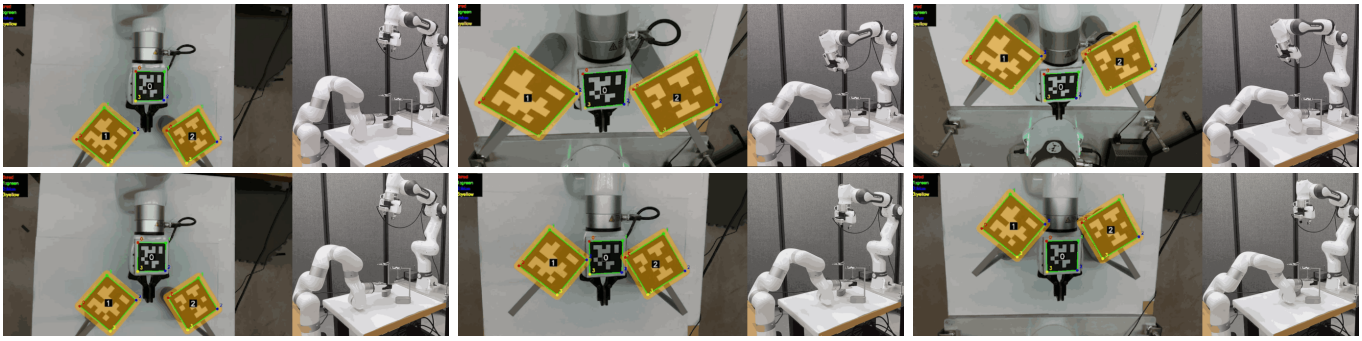
[22] G. M. Ziegler, *Lectures on Polytopes*. Springer Science & Business Media, 2012, vol. 152.

[23] 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, 2017.

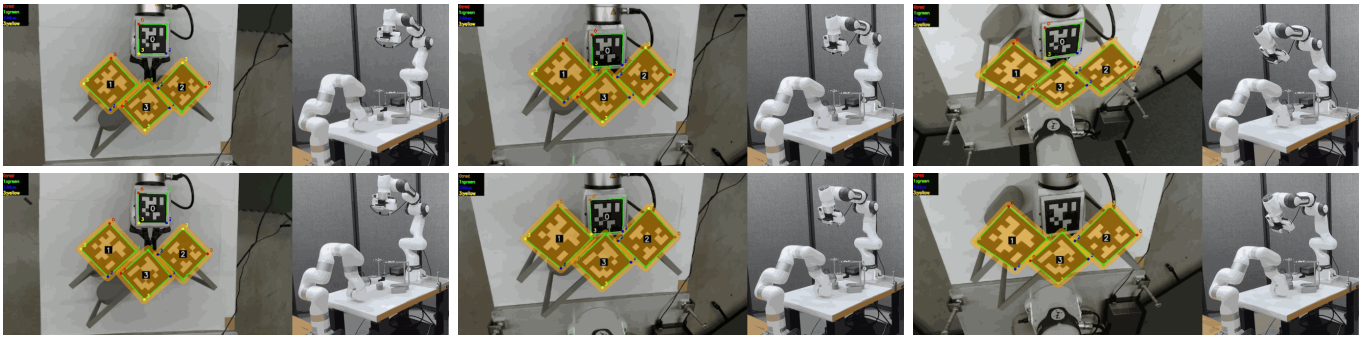
[24] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley: New York, 2006.



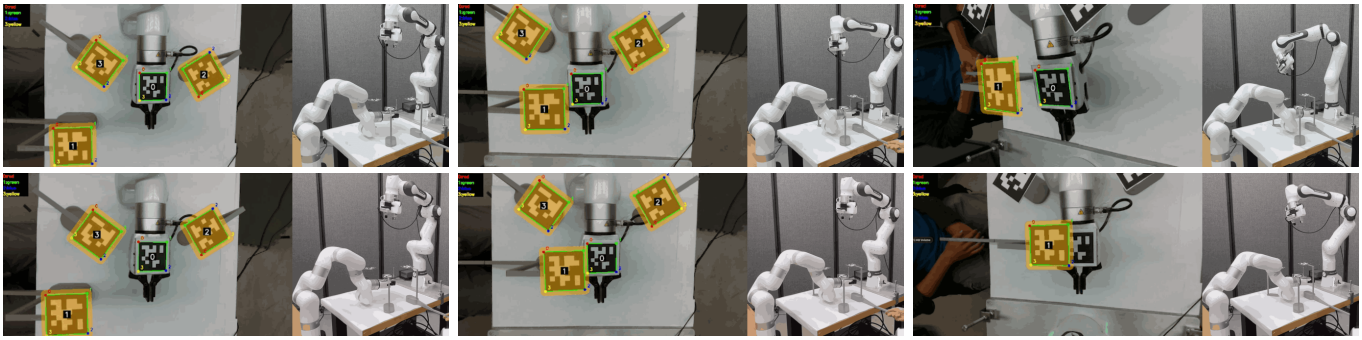
(a) Scenario 1 with one static occluding object: hierarchical controller (top row) versus MPC alone (bottom row).



(b) Scenario 2 with two static occluding objects: hierarchical controller (top row) versus MPC alone (bottom row)

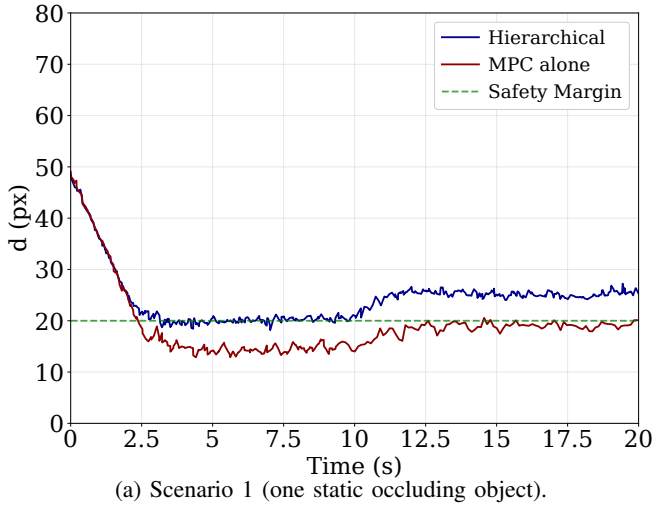


(c) Scenario 3 with three static occluding objects: hierarchical controller (top row) versus MPC alone (bottom row)

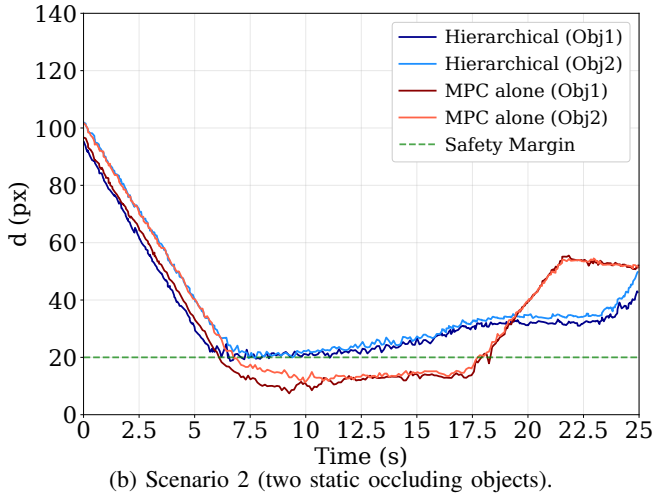


(d) Scenario 4 with one dynamic occluding objects: hierarchical controller (top row) versus MPC alone (bottom row)

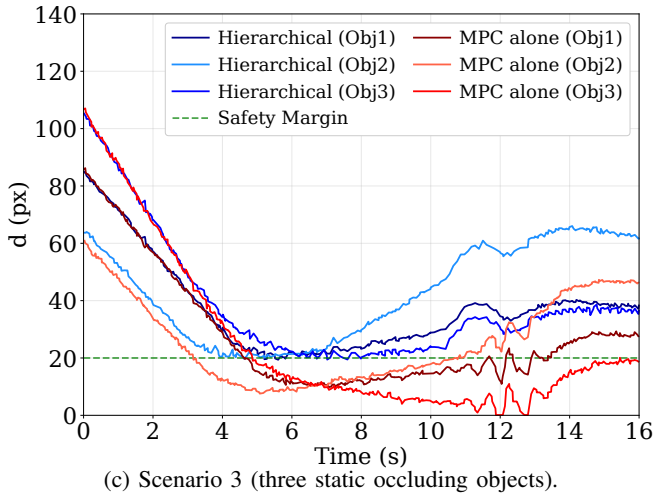
Figure 5: Snapshots of experiments for static and dynamic occluding objects. In each scenario, the top row displays the proposed hierarchical controller, which includes a high-level MPC and a low-level CBF-QP controller, while the bottom row shows the high-level MPC alone. In each row, the initial (left), intermediate (middle), and terminal (right) phases are shown from both a camera and a third-person view. Across all experiments, a UFactory manipulator with an AprilTag (ID 0) on its end-effector follows a linear trajectory toward a Franka manipulator. Occlusion zones are highlighted by yellow squares with a 20-pixel safety margin. AprilTag centers are annotated with their IDs, and tag corners are numbered clockwise from 0 to 3. The target (AprilTag 0) is enclosed in a green box when it is persistently visible, with the absence of the box indicating a loss of recognition due to occlusion.



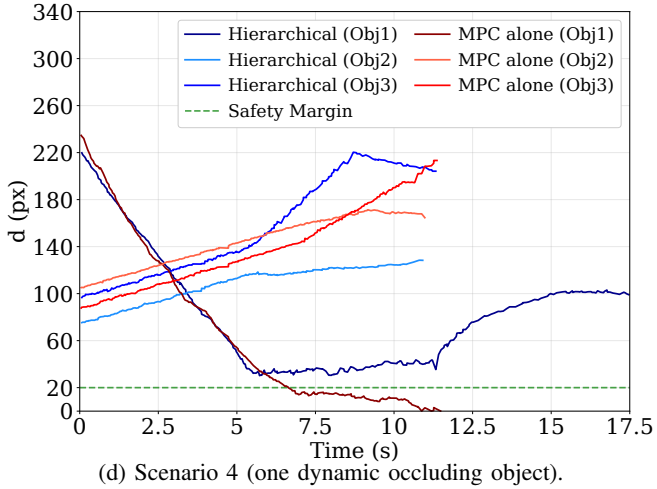
(a) Scenario 1 (one static occluding object).



(b) Scenario 2 (two static occluding objects).



(c) Scenario 3 (three static occluding objects).



(d) Scenario 4 (one dynamic occluding object).

Figure 6: Evolution of the distance between the target and the occluding objects across four experimental scenarios. In the legend, "Obj1" denotes $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_1)$, etc. In the static scenarios shown in (a)-(c), the high-level MPC alone fails to maintain the occlusion-avoidance constraint $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_j) \geq 20$ pixels, as the minimum distance drops below 20 for some occluding objects. In the dynamic scenario shown in (d), MPC alone allows $\mathrm{d}(\bar{\mathbb{T}}, \bar{\mathbb{O}}_1)$ to decrease to 20 at $t \approx 6.6$ s and drop to zero at $t \approx 11.5$ s, resulting in persistent occlusion and target recognition failure. Objects 2 and 3 exit the camera's field of view at $t \approx 11$ s for both controllers. In contrast, the hierarchical controller strictly enforces the occlusion-avoidance constraints in all scenarios, keeping distances near or above the 20-pixel safety margin, with only minor single-pixel deviations due to measure noise or hardware disturbances.

[25] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 972–983, 2020.

[26] W.-H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-observer-based control and related methods—An overview," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2015.

[27] Y. Wang and X. Xu, "Disturbance observer-based robust control barrier functions," in *American Control Conference*, 2023, pp. 2681–3687.

[28] B. Amos and J. Z. Kolter, "OptNet: Differentiable optimization as a layer in neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 136–145.

[29] P. Glotfelter, J. Cortés, and M. Egerstedt, "A nonsmooth approach to controller synthesis for boolean specifications," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5160–5174, 2020.

[30] A. Thirugnanam, J. Zeng, and K. Sreenath, "Duality-based convex optimization for real-time obstacle avoidance between polytopes with control barrier functions," in *American Control Conference*. IEEE, 2022, pp. 2239–2246.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[32] W. S. Cortez, D. Oetomo, C. Manzie, and P. Choong, "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 2, pp. 530–545, 2019.

[33] J. Breeden, K. Garg, and D. Panagou, "Control barrier functions in sampled-data systems," *IEEE Control Systems Letters*, vol. 6, pp. 367–372, 2021.

[34] Y. Zhang, S. Walters, and X. Xu, "Control barrier function meets interval analysis: Safety-critical control with measurement and actuation uncertainties," in *American Control Conference*, 2022, pp. 3814–3819.

[35] V. Freire and X. Xu, "Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2319–2334, 2023.

[36] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[37] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.