Adaptive Visual Servoing Control Barrier Function of Robotic Manipulators with Uncalibrated Camera

Jianing Zhao, Mingyang Feng, Yuepeng Zhang, Siqi Wang and Xiang Yin

Abstract— **This paper investigates the problem of safe visual servoing control of manipulators using an uncalibrated eye-in-hand camera based on control barrier functions (CBFs). Traditional CBFs are defined in the workspace, corresponding to the global coordinates of the base frame. However, when the camera's position or orientation is adjusted for a better field of view, it becomes uncalibrated, making it challenging to obtain the precise positions of the robot and obstacles using onboard sensors like a camera. To address this, we propose a novel *visual servoing control barrier function* (VS-CBF) for manipulators, which depends only on the image and depth data sensed by an RGB-D camera. Given an uncalibrated camera, we develop an adaptive estimator for the unknown camera parameters. Based on this estimator, we also design a kinematic visual servoing control law as a nominal controller, ensuring the convergence of the robotic system. The safe controller is then obtained by solving a quadratic programming problem that incorporates the designed VS-CBF and the nominal controller. Finally, experimental results conducted on a UR3 manipulator are presented to demonstrate the effectiveness of our approach.**

I. Introduction

Safety is one of the fundamental challenges in robotics and autonomous systems. Generally, safety refers to the requirement that an autonomous system never violates a given specification [19]. Since safety must be ensured over the entire operational horizon, safety-critical control has become a significant and challenging research topic. Over the past years, control barrier functions (CBFs) [5], [39], [4] have emerged as a powerful approach for designing controllers with provable safety guarantees. By defining a barrier function that encodes safety constraints, CBFs enable the synthesis of control policies that keep the system within a safe set at all times, even in the presence of uncertainties and disturbances.

CBFs were first introduced in [5] to enforce safety by defining a forward-invariant set characterized by a continuously differentiable function and an inequality constraint on the control input. Building on this foundation, various advanced CBF formulations have been developed to accommodate different classes of control systems. To be specific, a high-order CBF was proposed in [34] for systems with high relative degrees. For the similar purpose, the authors of [26], [25], [7] introduced a CBF construction approach for complex autonomous systems via reduced-order models. In [35], an adaptive CBF was designed to address systems with noisy dynamics and time-varying control bounds. Additionally, [8] considered CBFs tailored for Euler-Lagrange systems with input constraints.

Recently, the application of CBFs in robotics has demonstrated considerable success [9]. In [10] and [29], CBFs were employed to ensure field-of-view coverage and obstacle avoidance for UAVs, respectively. In [11], they were applied to legged robots to achieve safe foot placement. Moreover, CBFs have proven to be effective in enhancing safety in autonomous driving [36], [40]. The authors of [18] utilized CBFs to prevent singularities in robotic manipulators, while [37] explored their use in soft robots for compliant obstacle avoidance. In addition, an open-source software for implementing CBF-based safety in wheeled robots was developed in [6], demonstrating the effectiveness of CBFs on a human-support robot. Furthermore, in many scenarios, designing a specific CBF formulation is challenging due to the complex environments and unknown system dynamics. To address this, various learning-based approaches for synthesizing CBFs have been extensively studied in recent years [28], [27], [22], [20], [21].

On the other hand, constructing a feasible CBF generally relies on the global coordinate frame and precise state information. However, in many practical scenarios, obtaining exact state information is challenging due to sensor limitations and the absence of a common coordinate framework. Motivated by this, an increasing number of studies have explored CBFs for partially observed control systems and perception-based CBFs defined by various types of sensors. Specifically, in [2], [3], the authors proposed observer-based CBFs using output feedback to ensure the safety of partially observed control systems. In [42] and [14], image-based CBFs were designed to maintain the field of view for UAVs and to ensure connectivity in multi-UAV systems, respectively. In [1] and [15], image-based and LiDAR-based CBFs were introduced for obstacle avoidance in unknown environments. Furthermore, [41] developed a neural graph CBF leveraging LiDAR point clouds, demonstrating its effectiveness for obstacle avoidance in large-scale multi-robot systems. Despite these advancements, to the best of our knowledge, the problem of perception-based safe control for robotic manipulators with CBF-based safety guarantees remains an open challenge in the current literature.

Visual servoing control is one of the most important approaches for enabling robotic manipulators to perform

This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61803259).

All the authors are with the Department of Automation, Shanghai Jiao Tong University, and the Key Laboratory of System Control and Information Processing, the Ministry of Education of China, Shanghai 200240, China. E-mail: {jnzhao,Fmy-135214, singmal, sq_wang, yinxiang}@sjtu.edu.cn.

complex tasks using real-time vision feedback from cameras [17]. In many practical scenarios, cameras are uncalibrated, which can occur easily due to adjustments in their positions or orientations. To address this issue, [23] introduced the concept of a depth-independent interaction matrix, which ensures that unknown camera parameters appear linearly in the closed-loop dynamics. Based on this, the authors proposed an estimator to adaptively estimate these parameters and developed a visual servoing dynamic control law using the image error and estimated parameters to ensure the manipulator's convergence. Hereafter, the visual servoing problems of manipulators with uncalibrated fixed camera and eye-in-hand camera were successfully solved by [31], [32], [30]. Moreover, such visual servoing approach has also demonstrated effectiveness across various types of robotic manipulators, including the manipulation of deformable objects [12], [13], soft robotic manipulators [33], [38], [37], and flexible-joint manipulators [24]. However, the safety was not considered in most of the aforementioned works. Notably, the authors of [37] utilized CBFs to enforce force input constraints, though the CBF was not constructed in the image space. To this end, we are highly motivated to consider the general safety-critical visual servoing control problem of manipulators with uncalibrated camera.

In this paper, we investigate the problem of safe visual servoing control for manipulators equipped with an eye-in-hand RGB-D camera. We assume that the camera's position can be adjusted within a certain range, and thus its parameters are bounded by known upper and lower limits. First, we develop a kinematic servoing control law with an adaptive parameter estimator based on the image and depth information captured by the camera. This ensures the convergence of the manipulator and serves as a *nominal controller* in the subsequent safe control synthesis. Then, we introduce a new concept of *visual servoing control barrier function* (VS-CBF) defined in the space of image and depth, which can be dynamically updated in real time based on the sensed visual data. The safe controller is then synthesized by solving a quadratic programming problem that incorporates the designed VS-CBF with the nominal controller. Experiments on UR3 manipulators are conducted to validate the effectiveness of the proposed approach. The key contributions of this work are two-fold:

- We develop a nominal visual servoing kinematic controller for manipulators, in contrast to previous works that focus on dynamic-level control [23], [31], [32], [30]. This kinematic approach is more suitable for controlling physical manipulators, as their low-level dynamic control is typically well-designed and encapsulated, making it generally impossible to directly perform the dynamic control.
- Compared with the visual CBF proposed in [1] for mobile robots, the VS-CBF we propose is typically designed for the kinematics of manipulators and accounts for the uncalibrated camera by designing an adaptive parameter estimator, making it more effective for real-world manipulator control scenarios.

The remaining part of this paper is organized as follows. In Section II, the preliminaries are introduced and the CBF-based safe vision servoing kinematic control problem is formally defined. In Section III, we develop the vision servoing controller together with an adaptive estimator to achieve the image-based control at the kinematic level. In Section IV, we first propose the general definition of the visual servoing control barrier function (VS-CBF), based on which we design a specific VS-CBF and obtain the safe controller by solving a quadratic programming problem optimization problem that guarantees the safety. In Section V, experimental results are presented and the conclusion is drawn in Section VI.

Notations: For two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n$, we denote by $\mathbf{v}_1 \leq \mathbf{v}_2$ if $\mathbf{v}_1[i] \leq \mathbf{v}_2[i]$ for each $i = 1, \ldots, n$. given a vector $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}} \in \mathbb{R}^3$, we define its skew-symmetric matrix as

$$sk(\mathbf{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to be a *class $\mathcal{K}_\infty$ function*, denoted by $\alpha \in \mathcal{K}_\infty$, if $\alpha(0) = 0$, α is strictly increasing, and $\lim_{s\to\infty} \alpha(s) = \infty$. Furthermore, α is said to be an *extended class $\mathcal{K}_\infty$ function*, denoted by $\alpha \in \mathcal{K}_\infty^{e}$, if $\alpha(0) = 0$, α is strictly increasing, and $\lim_{s\to\pm\infty} \alpha(s) = \pm\infty$.

II. Preliminaries & Problem Formulation

A. Kinematics & Camera Model

Let $\boldsymbol{\xi}(t) \in \mathbb{R}^3$ and $\boldsymbol{\psi}(t) \in [-\pi, \pi]^3$ be the position and the orientation of the end-effector with respect to the robot base frame, respectively. Let $\mathbf{q}(t) \in \mathbb{R}^n$ be the joint angle of the manipulator, where n is the number of DOFs. The kinematics of the manipulator is given by

$$\begin{bmatrix} \dot{\boldsymbol{\xi}}(t) \\ \dot{\boldsymbol{\psi}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{J}_\xi(\mathbf{q}(t)) \\ \mathbf{J}_\psi(\mathbf{q}(t)) \end{bmatrix}}_{\mathbf{J}(\mathbf{q}(t))} \dot{\mathbf{q}}(t), \tag{1}$$

where $\mathbf{J}(\mathbf{q}(t)) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix of the manipulator. In the kinematic-level control of the robot, we will consider $\dot{\mathbf{q}}(t)$ as the control input, i.e.,

$$\mathbf{u}(t) = \dot{\mathbf{q}}(t). \tag{2}$$

In this paper, we focus on the position of end-effector whose dynamics is described by the following subsystem of (1)

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{J}_\xi(\mathbf{q}(t))\mathbf{u}(t). \tag{3}$$

The robot is equipped with a camera. Let ${}^b\mathbf{x} \in \mathbb{R}^3$ be the position of a fixed feature point with respect to the robot base frame and $\mathbf{y}(t) = [u(t), v(t)]^{\mathrm{T}}$ be the real-time position of the feature point in the image. Let $\mathbf{T}_e(t) \in \mathbb{R}^{4\times 4}$ be the homogenous transform matrix of the end-effector with respect to the robot base frame, respectively, i.e.,

$$\mathbf{T}_e^{-1}(t) = \begin{bmatrix} \mathbf{R}(t) & \boldsymbol{\xi}(t) \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where $\mathbf{R}(t)$ and $\boldsymbol{\xi}(t)$ are the rotation matrix and the translation vector from the robot base frame to the end-effector frame. The observation model of the camera is given by:

$$\begin{bmatrix}\mathbf{y}(t)\\1\end{bmatrix}=\frac{1}{{}^{c}z(t)}\mathbf{M}\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix}, \tag{5}$$

where $\mathbf{M}\in\mathbb{R}^{3\times4}$ is the perspective projection matrix, determined by the intrinsic and extrinsic parameters of the camera, and ${}^{c}z(t)$ is the depth of the feature point satisfying

$${}^{c}z(t)=\mathbf{m}_3^{\mathrm{T}}\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix} \tag{6}$$

where we use $\mathbf{m}_i^{\mathrm{T}}$ to denote the i-th row of $\mathbf{M}$ for $i=1,2,3$.

To describe the dynamics of the image and the control input, we take the time derivative of (5), which yields,

$$\dot{\mathbf{y}}(t)=\frac{1}{{}^{c}z(t)}\mathbf{A}(t)\dot{\mathbf{q}}(t), \tag{7}$$

where

$$\mathbf{A}(t)=\begin{bmatrix}\mathbf{m}_1^{\mathrm{T}}-u(t)\mathbf{m}_3^{T}\\ \mathbf{m}_2^{\mathrm{T}}-v(t)\mathbf{m}_3^{T}\end{bmatrix}\frac{\partial\left(\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix}\right)}{\partial\mathbf{q}} \tag{8}$$

is called the *depth-independent interaction matrix* or the image Jacobian matrix [32], and

$$\begin{aligned}&\frac{\partial\left(\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix}\right)}{\partial\mathbf{q}}=\frac{\partial\left(\mathbf{R}(t){}^{b}\mathbf{x}+\boldsymbol{\xi}(t)\right)}{\partial\mathbf{q}}\\&=\begin{bmatrix}-\mathbf{I}_{3\times3} & sk(\mathbf{R}(t){}^{b}\mathbf{x}+\boldsymbol{\xi}(t))\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3}\end{bmatrix}\begin{bmatrix}\mathbf{R}(t) & 0\\ 0 & \mathbf{R}(t)\end{bmatrix}\mathbf{J}(\mathbf{q}(t)).\end{aligned}$$

Furthermore, we take the derivative of (6), yielding

$${}^{c}\dot{z}(t)=\frac{\partial\left(\mathbf{m}_3^{\mathrm{T}}\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix}\right)}{\partial\mathbf{q}}\dot{\mathbf{q}}(t)=:\mathbf{a}^{\mathrm{T}}(t)\dot{\mathbf{q}}(t) \tag{9}$$

where $\mathbf{a}^{\mathrm{T}}(t)$ is a vector determined by the camera parameters and the joint angles of the manipulator.

However, in many situations, the perspective projection matrix $\mathbf{M}$ is unknown due to the unavailable intrinsic and extrinsic parameters of the camera. This is typically common in scenarios where the position and the orientation of the camera need to be adjusted *within a certain range* for a better field of view before implementing the controller. Therefore, we have to estimate $\mathbf{M}$ adaptively together with the control law. Also, ${}^{b}\mathbf{x}$ is also unavailable in most of the scenarios when the camera is uncalibrated. Given the feature point in the image $\mathbf{y}(t)$, the depth ${}^{c}z(t)$ as well as the transformation matrix $\mathbf{T}_e(t)$, to express (5) in a way of linear form of parameters, we denote the perception projection matrix by $\mathbf{M}=\begin{bmatrix}\boldsymbol{\Omega} & \boldsymbol{\chi}\end{bmatrix}$, where $\boldsymbol{\Omega}\in\mathbb{R}^{3\times3}$ and $\boldsymbol{\chi}\in\mathbb{R}^{3}$ is the left matrix and the right vector, respectively. Then we re-write (5) with (4) as

$$\begin{aligned}{}^{c}z(t)\begin{bmatrix}\mathbf{y}(t)\\1\end{bmatrix}&=\mathbf{M}\mathbf{T}_e^{-1}(t)\begin{bmatrix}{}^{b}\mathbf{x}\\1\end{bmatrix}\\&=\begin{bmatrix}\boldsymbol{\Omega} & \boldsymbol{\chi}\end{bmatrix}\begin{bmatrix}\mathbf{R}(t){}^{b}\mathbf{x}+\boldsymbol{\xi}(t)\\1\end{bmatrix}\\&=\boldsymbol{\Omega}\mathbf{R}(t){}^{b}\mathbf{x}+\boldsymbol{\Omega}\boldsymbol{\xi}(t)+\boldsymbol{\chi}\\&=\boldsymbol{\Phi}(\mathbf{q}(t))\boldsymbol{\theta}_p,\end{aligned} \tag{10}$$

where $\boldsymbol{\theta}_p\in\mathbb{R}^{39}$ is the vector of parameters that decide both $\mathbf{M}$ and ${}^{b}\mathbf{x}$, and $\boldsymbol{\Phi}(\mathbf{q}(t))\in\mathbb{R}^{3\times39}$ is a matrix decided by the motion of the end-effector. The reader is referred to [32] for more details on the above property of matrix $\mathbf{M}$.

Due to the uncalibrated camera, in what follows, we use $\hat{\mathbf{M}}(t)$, $\hat{\mathbf{A}}(t)$, $\hat{\mathbf{a}}(t)$, ${}^{b}\hat{\mathbf{x}}(t)$ and $\hat{\boldsymbol{\theta}}_p(t)$ to denote the estimated value to $\mathbf{M}$, $\mathbf{A}(t)$, $\mathbf{a}(t)$, ${}^{b}\mathbf{x}$ and $\boldsymbol{\theta}_p$, respectively.

Furthermore, since the camera is adjusted within a certain range, we make the following assumptions.

Assumption 1: All the unknown parameters of the camera are are bounded by known upper and lower bounds.

Based on the above assumption, we have $\underline{\boldsymbol{\theta}}_p\le\boldsymbol{\theta}_p\le\overline{\boldsymbol{\theta}}_p$, where $\underline{\boldsymbol{\theta}}_p$ and $\overline{\boldsymbol{\theta}}_p$ are known.

Also, to achieve continuous kinematic control, we make the following assumption based on the camera's field of view.

Assumption 2: The feature point is in the field of view of the camera all the time.

B. Safety & Control Barrier Function

In this paper, we only take into consideration the safety of the end-effector in the workspace, i.e., the safety of $\boldsymbol{\xi}(t)$, which is without loss of generality due to the fact we can still consider the whole body of the robotic manipulator by defining an augmented state incorporating its configuration.

Let $\mathcal{X}_S\subset\mathbb{R}^3$ be the set of all safe states. Then we define the safety of manipulator as follows.

Definition 1 (Safety): System (3) is considered to be safe if $\boldsymbol{\xi}(t)\in\mathcal{X}_S$ for all $t\ge t_0$.

To ensure the safety of system (3) over a suitable safe set, we introduce the definition of *forward invariant set*.

Definition 2 (Forward Invariance): A set $\mathcal{C}\subset\mathbb{R}^3$ is said to be forward invariant for system (3) if for any $\boldsymbol{\xi}(t_0)\in\mathcal{C}$, we have $\boldsymbol{\xi}(t)\in\mathcal{C}$ for any $t\ge t_0$.

In this paper, we aim to use a continuously differentiable function $h:\mathbb{R}^3\to\mathbb{R}$ to characterize a safe set $\mathcal{C}$ that is forward invariant for the system (3). To this end, given a continuously differentiable function $h:\mathbb{R}^3\to\mathbb{R}$, we focus on the following sets

$$\mathcal{C}=\{\boldsymbol{\xi}\in\mathbb{R}^3:h(\boldsymbol{\xi})\ge0\} \tag{11a}$$

$$\partial\mathcal{C}=\{\boldsymbol{\xi}\in\mathbb{R}^3:h(\boldsymbol{\xi})=0\} \tag{11b}$$

$$\mathrm{Int}(\mathcal{C})=\{\boldsymbol{\xi}\in\mathbb{R}^3:h(\boldsymbol{\xi})>0\} \tag{11c}$$

where $\mathcal{C}$ is the zero-superlevel of h, $\partial\mathcal{C}$ is the set boundary, and Int$(\mathcal{C})$ is the interior of set $\mathcal{C}$. Then, we introduce the standard control barrier function as follows, which guarantees that its zero-superlevel set is a forward invariant set.

Definition 3 (Control Barrier Function): A continuously differentiable function $h:\mathbb{R}^3\to\mathbb{R}$ defining a set $\mathcal{C}\subset\mathbf{R}^3$

as in (11a) is said to be a *control barrier function* (CBF) for system (3) on $\mathcal{C}$ if there exists a function $\alpha \in \mathcal{K}_\infty^e$ such that for all $\boldsymbol{\xi} \in \mathbb{R}^3$, we have

$$\sup_{\mathbf{u}(t)\in\mathbb{R}^n} \frac{\partial h(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \mathbf{J}_\xi(\mathbf{q}(t))\mathbf{u}(t) \geq -\alpha(h(\boldsymbol{\xi})). \tag{12}$$

Here, we claim that, the above condition can always be satisfied by setting a suitable function α, due to the fact that system (3) has relative degree 1, i.e., $\frac{\partial h(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \mathbf{J}_\xi(\mathbf{q}(t)) \neq \mathbf{0}$, and the control input $\mathbf{u}(t)$ can be chosen in $\mathbb{R}^n$.

C. Problem Formulation

In general, the above standard CBF $h : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined over the workspace $\mathbb{R}^3$, whose construction requires both of the exact positions of obstacles and the end-effector in the robot base frame. However, despite the availability of the end-effector's position from the manipulator, the uncalibrated camera makes it challenging to accurately determine the position of the obstacles. Therefore, in this paper, we aim to directly construct a CBF over the image space together with the depth that is available from an uncalibrated RGB-D camera. Apart from the safety, due to the uncalibrated camera, it is also necessary to design a kinematic controller based on the image and the depth to achieve some goals for the manipulator as a *nominal controller*. In what follows, we provide the formal description for both the above problems.

For the nominal kinematic controller, we aim to accomplish the goal of manipulator by designing a controller that guarantees the position of the feature point on the image converging to a desired position. Let $\mathbf{y}_d$ be the desired position of the feature point on image and define the image error of the feature point as

$$\Delta\mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}_d. \tag{13}$$

However, given the uncalibrated camera, it is necessary to adaptively estimate the unknown parameters based on the image of the feature point $\mathbf{y}(t_j)$ with depth ${}^c z(t_j)$ and the available transformation matrix $\mathbf{T}_e(t_j)$ in the history, where $t_j \leq t$ is the sampling time instant. In general, to estimate the perception projection matrix $\mathbf{M}$ and the position of a feature point ${}^b\mathbf{x}$, there are 39 parameters required in (10), which requires at least 20 images with $\mathbf{y}(t_j)$ and ${}^c z(t_j)$ for $j = 1, \ldots, 20$.

Then, we formulate the vision servoing control problem with an uncalibrated camera as follows.

Problem 1 (Adaptive Vision Servoing Kinematic Control): Consider the manipulator described by system (1) with an uncalibrated camera satisfying Assumption 1 and 2. Design a nominal controller of the following form

$$\hat{\mathbf{u}}(t) = \boldsymbol{\rho}(\mathbf{y}(t), \mathbf{y}_d, \hat{\boldsymbol{\theta}}_p(t)), \tag{14a}$$

$$\dot{\hat{\boldsymbol{\theta}}}_p(t) = \boldsymbol{\varrho}(\hat{\boldsymbol{\theta}}_p(t), \{\mathbf{y}(t_j), {}^c z(t_j), \mathbf{T}_e(t_j)\}_{t_j \leq t}), \tag{14b}$$

such that

$$\lim_{t\rightarrow\infty} \Delta\mathbf{y}(t) = 0, \quad \lim_{t\rightarrow\infty} \hat{\boldsymbol{\theta}}_p(t) = l\boldsymbol{\theta}_p, \tag{15}$$

where l is a constant decided by one component of $\mathbf{M}$, and $\boldsymbol{\rho}$ and $\boldsymbol{\varrho}$ are sufficiently smooth functions to be designed.

Remark 1: In theory, we are only able to estimate the matrix $\mathbf{M}$ up to a scale due to the following fact

$$\begin{bmatrix} \mathbf{y}(t) \\ 1 \end{bmatrix} = \frac{l\mathbf{M}}{l\,{}^c z(t)} \mathbf{T}_e^{-1}(t) \begin{bmatrix} {}^b\mathbf{x} \\ 1 \end{bmatrix}. \tag{16}$$

Therefore, it is sufficient to fix one component of $\boldsymbol{\theta}_p$ as 1 and then $\hat{\boldsymbol{\theta}}_p(t)$ is expected to converge to $\boldsymbol{\theta}_p$ up to scale l.

With the nominal kinematic controller in (14), we now take the safety into consideration. The safe vision servoing control of manipulators is defined as follows.

Problem 2 (Safe Vision Servoing Kinematic Control): Consider a workspace of the manipulator in $\mathbb{R}^3$ where there are obstacles $\mathcal{O} \subset \mathbb{R}^3$ that are unknown to the manipulator. The safe region avoiding the obstacles is captured by the safety set $\mathcal{X}_S = \mathbb{R}^3 \backslash \mathcal{O}$. The kinematics of the manipulator is described by (1) with input (2). The obstacles and the end-effector are sensed by an uncalibrated camera satisfying Assumption 1 and 2 through (5) as $\mathbf{y}_\mathcal{O}(t)$ with ${}^c z_\mathcal{O}(t)$ and $\mathbf{y}_\xi$, respectively. Let $\hat{\boldsymbol{\theta}}_\mathcal{O}(t)$ be the estimated parameter by choosing the obstacle as the feature point. Design a control barrier function in the following form of

$$h(t) = h(\mathbf{y}_\mathcal{O}(t), \mathbf{y}_\xi, {}^c z_\mathcal{O}(t)) \tag{17}$$

with a constraint condition

$$\varphi(h(\mathbf{y}_\mathcal{O}(t), \mathbf{y}_\xi, {}^c z_\mathcal{O}(t)), \hat{\boldsymbol{\theta}}_\mathcal{O}(t), \mathbf{u}(t)) \geq 0 \tag{18}$$

such that the zero-superlevel defined by $h(t)$ is a safety set, i.e.,

$$\mathcal{C}(t) = \{\boldsymbol{\xi}(t) \in \mathbb{R}^3 : h(t) \geq 0\} \subseteq \mathcal{X}_S \tag{19}$$

and the zero-superlevel set of $h(t)$ defined in is forward invariant for system (3), i.e.,

$$\boldsymbol{\xi}(t_0) \in \mathcal{C}(t_0) \;\Rightarrow\; \boldsymbol{\xi}(t) \in \mathcal{C}(t). \tag{20}$$

where φ is a sufficiently smooth function to be designed.

Remark 2: Here, we slightly abuse the notation $\mathbf{y}_\mathcal{O}(t)$ and ${}^c z_\mathcal{O}(t)$ to describe the positions of the obstacles on the image and their depths. In fact, since the obstacles $\mathcal{O}$ are objects with shapes and sizes, in the image space, $\mathbf{y}_\mathcal{O}(t)$ is generally a region denoting the pixels for the obstacles, instead of one single pixel. Also, each pixel in the region $\mathbf{y}_\mathcal{O}(t)$ is coupled with a depth for which we use ${}^c z_\mathcal{O}(t)$ to denote all depths returned by the RGB-D camera. Furthermore, the camera is fixed on the end of the manipulator (after necessary adjustment), so the end-effector position on the image should be a constant $\mathbf{y}_\xi$.

With the above two problems being solved, we can easily obtain a safe vision servoing kinematic controller that achieves the specific task and avoids obstacles by solving a quadratic programming with the nominal controller $\hat{\mathbf{u}}(t)$ and the visual CBF $h(t)$, which will be formally presented in the following sections.

III. Adaptive Visual Kinematic Controller

In this section, we propose a kinematic controller together with an adaptive parameter estimator that achieves both of the convergence of the feature point to a desired position on the image and the convergence of camera parameters based on the image and depth information returned by the uncalibrated camera.

Inspired by [32], we first introduce the following Frobenius norm of errors to evaluate the time-varying error of the estimated matrix $\hat{\mathbf{M}}$ and the estimated position of the feature point ${}^b\hat{\mathbf{x}}$, i.e.,

$$\mathbf{e}_p(t;t_j) = {}^c z(t_j)\mathbf{y}(t_j) - \begin{bmatrix}\hat{\mathbf{m}}_1^{\mathrm{T}}(t)\\ \hat{\mathbf{m}}_2^{\mathrm{T}}(t)\end{bmatrix} \mathbf{T}_e^{-1}(t_j) \begin{bmatrix}{}^b\hat{\mathbf{x}}(t)\\ 1\end{bmatrix}. \quad (21)$$

From (5), we have

$$ {}^c z(t_j)\mathbf{y}(t_j) - \begin{bmatrix}\mathbf{m}_1^{\mathrm{T}}\\ \mathbf{m}_2^{\mathrm{T}}\end{bmatrix} \mathbf{T}_e^{-1}(t_j) \begin{bmatrix}{}^b\mathbf{x}\\ 1\end{bmatrix} = 0. \quad (22)$$

Combining (21) and (22), we have

$$\mathbf{e}_p(t;t_j) = \begin{bmatrix}\mathbf{m}_1^{\mathrm{T}}\\ \mathbf{m}_2^{\mathrm{T}}\end{bmatrix} \mathbf{T}_e^{-1}(t_j) \begin{bmatrix}{}^b\mathbf{x}\\ 1\end{bmatrix} - \begin{bmatrix}\hat{\mathbf{m}}_1^{\mathrm{T}}(t)\\ \hat{\mathbf{m}}_2^{\mathrm{T}}(t)\end{bmatrix} \mathbf{T}_e^{-1}(t_j) \begin{bmatrix}{}^b\hat{\mathbf{x}}(t)\\ 1\end{bmatrix}.$$

From (10), we have

$$\begin{bmatrix}\mathbf{e}_p(t;t_j)\\ 1\end{bmatrix} = \mathbf{\Phi}(\mathbf{q}(t_j))(\boldsymbol{\theta}_p - \hat{\boldsymbol{\theta}}_p(t)). \quad (23)$$

By denoting $\Delta\boldsymbol{\theta}_p = \hat{\boldsymbol{\theta}}_p(t) - \boldsymbol{\theta}_p$, it is easy to find a constant matrix $\mathbf{W}_p(t_j) \in \mathbf{R}^{2\times 39}$ such that

$$\mathbf{e}_p(t;t_j) = \mathbf{W}_p(t_j)\Delta\boldsymbol{\theta}_p(t). \quad (24)$$

Given an estimated $\hat{\mathbf{M}}(t)$, we can compute the estimated depth-independent interaction matrix $\hat{\mathbf{A}}(t)$ by replacing $\mathbf{M}$ by $\hat{\mathbf{M}}(t)$ in (8). Next, we analyze the property of the error of the estimated $\hat{\mathbf{A}}(t)$, i.e., $\hat{\mathbf{A}}(t) - \mathbf{A}(t)$.

Based on the result of [32], the depth-independent interaction matrix $\mathbf{A}(t)$ has the following property: for any vector $\mathbf{v} \in \mathbb{R}^n$, the product $\mathbf{A}(t)\mathbf{v}$ can be written as a linear form of the unknown parameters $\boldsymbol{\theta}_p$ with a constant $\mathbf{w}_p \in \mathbb{R}^2$, i.e.,

$$\mathbf{A}(t)\mathbf{v} = \mathbf{Y}(\mathbf{v},\mathbf{y}(t))\boldsymbol{\theta}_p + \mathbf{w}_p \quad (25)$$

where $\mathbf{Y}(\mathbf{v},\mathbf{y}(t)) \in \mathbb{R}^{2\times n}$ is a regressor matrix that does not depend on the parameters. Therefore, we have

$$(\hat{\mathbf{A}}(t) - \mathbf{A}(t))\dot{\mathbf{q}}(t) = \mathbf{Y}_p(\mathbf{q}(t),\mathbf{y}(t))\Delta\boldsymbol{\theta}_p(t) \quad (26)$$

With the above results, we propose the kinematic controller and the adaptive estimator as follows:

$$\hat{\mathbf{u}}(t) = -\hat{\mathbf{A}}^{\mathrm{T}}(t)\mathbf{B}\Delta\mathbf{y}(t) \quad (27a)$$

$$\dot{\hat{\boldsymbol{\theta}}}_p(t) = -\frac{1}{{}^c z(t)}\mathbf{\Gamma}^{-1}\mathbf{Y}_p^{\mathrm{T}}(\mathbf{q}(t),\mathbf{y}(t))\mathbf{B}\Delta\mathbf{y}(t) - \mathbf{\Gamma}^{-1}\sum_{j=1}^{k}\mathbf{W}_p^{\mathrm{T}}(t_j)\mathbf{K}\mathbf{e}_p(t_j;t) \quad (27b)$$

where $k \geq 20$ is the sampling times, and $\mathbf{B}\in\mathbb{R}^{2\times 2}$, $\mathbf{\Gamma} \in \mathbb{R}^{39\times 39}$ and $\mathbf{K}\in\mathbb{R}^{2\times 2}$ are positive definite matrices.

The intuition behind the design of the above control law is explained as follows. In the controller, we use $\hat{\mathbf{A}}^{\mathrm{T}}(t)\mathbf{B}\Delta\mathbf{y}(t)$ to control with the image error feedback. Note that the true value of $\mathbf{A}(t)$ is not involved and the estimated $\hat{\mathbf{A}}(t)$ can be easily obtained according to the real-time estimated parameters $\hat{\boldsymbol{\theta}}_p(t)$. In the design of the estimator, we use the first term to cancel the regressor term introduced by the estimation error $\hat{\mathbf{A}}(t) - \mathbf{A}(t)$ in the controller, and use the second term to ensure the convergence of the estimated parameter $\hat{\boldsymbol{\theta}}_p(t)$ by the feedback with the estimation error captured by $\mathbf{e}_p(t;t_j)$.

Then, the convergence of the above control law is summarized by the following theorem.

Theorem 1: If Assumptions 1 and 2 are satisfied, under the controller (27), the image error $\Delta\mathbf{y}(t)$ of system (1) converges to zero while the estimated parameters converge to the real values up to a scale, i.e.,

$$\lim_{t\to\infty}\Delta\mathbf{y}(t) = 0, \quad \lim_{t\to\infty}\hat{\boldsymbol{\theta}}_p(t) = l\boldsymbol{\theta}_p. \quad (28)$$

Proof: All the proofs can be found in the Appendix. ■

Remark 3: Compared to [32], we design the kinematic controller instead of the control law on the level of dynamics of the manipulator. As a matter of fact, for most of the manipulators, since the low-level dynamic control of manipulators has been well-designed and encapsulated, making it generally impossible to directly perform the dynamic control. Furthermore, compared to [32], we have incorporated more depth information in our controller design. This is because the depth can be easily obtained from an RGB-D camera. Moreover, unlike the work in [32], in addition to convergence tasks, we also need to consider obstacle avoidance. Therefore, depth information is essential for both the controller design and the following design of visual servoing CBF.

IV. Visual Servoing Control Barrier Function & Safe Visual Servoing Kinematic Control

In this section, we first introduce the control barrier function defined in the image-depth space. Then, we propose the safe visual servoing controller by solving a quadratic programming (QP) optimization problem.

A. Visual Servoing Control Barrier Function

In this subsection, we first introduce the general form of the visual servoing CBF defined over the image-depth space, and then design a specific CBF based on the image position of the obstacles $\mathbf{y}_O(t)$ together with the depths ${}^c z_{\mathcal{O}}(t)$ and the image position of the end-effector $\mathbf{y}_\xi$, which satisfies all the requirement of the visual servoing CBF.

In general, there are multiple obstacles in the workspace and we use $\mathbf{y}_{\mathcal{O}}(t)$ and ${}^c z_{\mathcal{O}}(t)$ to describe the pixels and the corresponding depths of all the obstacles. However, for the sake of writing convenience, in what follows, we assume that there is only *one obstacle*. We claim that such assumption is without loss of generality since our framework can be extended to the arbitrary number of obstacles.

Furthermore, with a slight abuse of the notation, in what follows, we use $\mathbf{y}_{\mathcal{O}}(t)$ and ${}^c z_{\mathcal{O}}(t)$ to denote the *center coordinates of the pixels of the obstacle* and the *minimum depth of the obstacle* at time t, respectively.

Now, we introduce the general visual servoing CBF candidate in the image and depth space in the following form of:

$$h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) = h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}) + h_{z}({}^{c}z_{\mathcal{O}}(t)). \quad (29)$$

Then, we define the safety set in the image and depth space induced by $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ as

$$\mathcal{S}(t) = \{(\mathbf{y}_{\mathcal{O}}(t), {}^{c}z(t)) : h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) \geq 0\}. \quad (30)$$

Also, we equivalently define the safety set in the workspace as

$$\mathcal{C}(t) = \{\boldsymbol{\xi}(t) \in \mathbb{R}^{3} : h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) \geq 0\}. \quad (31)$$

Naturally, we have

$$((\mathbf{y}_{\mathcal{O}}(t), {}^{c}z(t))) \in \mathcal{S}(t) \ \Rightarrow \ \boldsymbol{\xi}(t) \in \mathcal{C}(t). \quad (32)$$

Intuitively, we use $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi})$ to capture the collision avoidance on the image, where $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}) > 0$ indicates that the end-effector is safe on the image at time t. However, $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}) \leq 0$ does not necessarily mean that the end-effector collides with the obstacle in the workspace. To this end, we use $h_{z}({}^{c}z_{\mathcal{O}}(t))$ to capture the safety in the space of depth, for which $h_{z}({}^{c}z_{\mathcal{O}}(t)) > 0$ indicates that the distance between the end-effector and the obstacle is safe at time t. The reason why we use both $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi})$ and $h_{z}({}^{c}z_{\mathcal{O}}(t))$ to capture the safety, instead of using $h_{z}({}^{c}z_{\mathcal{O}}(t))$ only lies in that the existence of $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi})$ also reduces the conservativeness of the CBF. That is, if the condition $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}) > 0$ allows that the distance between the end-effector and the obstacle is not necessarily too large.

With the above form of visual servoing CBF candidate, we need to form a constraint condition on the derivative of $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ with the control input $\mathbf{u}(t)$ involved. It is obvious that the dynamics between $\mathbf{y}_{\mathcal{O}}(t)$ and $\mathbf{u}(t)$ satisfies (7) as well. However, since the camera is uncalibrated, the depth-independent interaction matrix with respect to the object $\mathbf{A}_{\mathcal{O}}(t)$ is unavailable. Similar to parameter estimation with respect to the feature point in Section III, we denote by $\boldsymbol{\theta}_{\mathcal{O}}$ the parameter that is related to the obstacle, which also satisfies

$$\underline{\boldsymbol{\theta}}_{\mathcal{O}} \leq \boldsymbol{\theta}_{\mathcal{O}} \leq \overline{\boldsymbol{\theta}}_{\mathcal{O}} \quad (33)$$

based on Assumption 1. Then, we propose the following estimation law for $\boldsymbol{\theta}_{\mathcal{O}}$:

$$\dot{\hat{\boldsymbol{\theta}}}_{\mathcal{O}}(t) = -\boldsymbol{\Gamma}^{-1} \sum_{j=1}^{k} \mathbf{W}_{\mathcal{O}}^{\mathrm{T}}(t_j)\mathbf{K}\mathbf{e}_{\mathcal{O}}(t_j; t), \quad (34)$$

where the initial condition satisfies $\underline{\boldsymbol{\theta}}_{\mathcal{O}} \leq \hat{\boldsymbol{\theta}}_{\mathcal{O}}(t_0) \leq \overline{\boldsymbol{\theta}}_{\mathcal{O}}$. By a similar stability analysis with Appendix I, we can easily conclude that $\hat{\boldsymbol{\theta}}_{\mathcal{O}}(t) \to l\boldsymbol{\theta}_{\mathcal{O}}$ as $t \to \infty$. We denote by $\hat{\mathbf{A}}_{\mathcal{O}}(t)$ and $\hat{\mathbf{a}}_{\mathcal{O}}(t)$ the estimated $\mathbf{A}_{\mathcal{O}}(t)$ and $\mathbf{a}_{\mathcal{O}}(t)$ from the estimated parameter $\hat{\boldsymbol{\theta}}_{\mathcal{O}}(t)$, respectively. Furthermore, similar to (26), the estimation errors $\hat{\mathbf{A}}_{\mathcal{O}}(t) - \mathbf{A}_{\mathcal{O}}(t)$ and $\hat{\mathbf{a}}_{\mathcal{O}}(t) - \mathbf{a}_{\mathcal{O}}(t)$ satisfy

$$(\hat{\mathbf{A}}_{\mathcal{O}}(t) - \mathbf{A}_{\mathcal{O}}(t))\dot{\mathbf{q}}(t) = \mathbf{Y}_{\mathcal{O}}(\mathbf{q}(t), \mathbf{y}(t))\Delta\boldsymbol{\theta}_{\mathcal{O}}(t), \quad (35a)$$

$$(\hat{\mathbf{a}}_{\mathcal{O}}(t) - \mathbf{a}_{\mathcal{O}}(t))\dot{\mathbf{q}}(t) = \mathbf{Y}_{z}(\mathbf{q}(t), \mathbf{y}(t))\Delta\boldsymbol{\theta}_{\mathcal{O}}(t), \quad (35b)$$

where $\mathbf{Y}_{\mathcal{O}} \in \mathbb{R}^{2\times n}$ and $\mathbf{Y}_{z}^{\mathrm{T}} \in \mathbb{R}^{n}$ are a regression matrix and a regression vector, respectively.

Now, we propose the formal definition of the visual servoing CBF with the constraint condition as follows:

Definition 4 (Visual Servoing CBF): A continously differentiable function $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ in the form of (29) is a *visual control barrier function* (VS-CBF) for system (3) if there exists a function $\alpha \in \mathcal{K}_{\infty}^{e}$ such that for all $(\mathbf{y}_{\mathcal{O}}(t), {}^{c}z(t)) \in \mathcal{S}(t)$, the following condition holds

$$\sup_{\mathbf{u}(t)\in\mathbb{R}^{n}} \left[\left(\frac{\partial h_{\mathbf{y}}}{\partial \mathbf{y}_{\mathcal{O}}} \frac{1}{{}^{c}z_{\mathcal{O}}(t)} \hat{\mathbf{A}}_{\mathcal{O}}(t) + \frac{\partial h_{z}}{\partial\, {}^{c}z_{\mathcal{O}}(t)} \hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t) \right) \mathbf{u}(t) - B(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) \right] \geq -\alpha(h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))), \quad (36)$$

where $\hat{\mathbf{A}}_{\mathcal{O}}(t)$ and $\hat{\mathbf{a}}(t)$ are computed from $\hat{\boldsymbol{\theta}}_{\mathcal{O}}(t)$ which is updated according to the estimation law (34) and

$$B(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) = \quad (37)$$
$$\left(\frac{1}{{}^{c}z_{\mathcal{O}}(t)} \left\| \frac{\partial h_{\mathbf{y}}}{\partial \mathbf{y}_{\mathcal{O}}} \right\| \|\mathbf{Y}_{\mathcal{O}}\| + \left\| \frac{\partial h_{z}}{\partial\, {}^{c}z_{\mathcal{O}}(t)} \right\| \|\mathbf{Y}_{z}\| \right) \left\| \overline{\boldsymbol{\theta}}_{\mathcal{O}} - \underline{\boldsymbol{\theta}}_{\mathcal{O}} \right\| .$$

Now, we establish the main result of the visual CBF as the following theorem.

Theorem 2: Consider the system (3) driven by a controller $\mathbf{u}(t)$ satisfying (36). Given the safety sets $\mathcal{S}(t)$ and $\mathcal{C}(t)$ defined by the VS-CBF $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ in (30) and in (31), respectively, under Assumption 1, we have

$$(\mathbf{y}_{\mathcal{O}}(t_0), {}^{c}z(t_0)) \in \mathcal{S}(t_0) \ \Rightarrow \ (\mathbf{y}_{\mathcal{O}}(t), {}^{c}z(t)) \in \mathcal{S}(t). \quad (38)$$

That is, $\mathcal{S}(t)$ is forward invariant. Equivalently, we have

$$\boldsymbol{\xi}(t_0) \in \mathcal{C}(t_0) \ \Rightarrow \ \boldsymbol{\xi}(t) \in \mathcal{C}(t). \quad (39)$$

Next, we aim to design a specific control barrier function that satisfies the condition (36) and ensures that

$$\mathcal{C}(t) \subseteq \mathcal{X}_{S}. \quad (40)$$

Here we propose a simple VS-CBF candidate as follows:

$$h(t) = \lambda \left((\mathbf{y}_{\mathcal{O}}(t) - \mathbf{y}_{\xi})^{2} - r_{\mathcal{O}}^{2} \right) + ({}^{c}z_{\mathcal{O}}(t) - d_{s}) \quad (41)$$

where $r_{\mathcal{O}} \in \mathbb{R}_{+}$ is an over-approximation for the radius of the obstacle pixels on the image, $d_s \in \mathbb{R}_{+}$ is a safety distance we set based on the size of the end-effector, and $\lambda \in \mathbb{R}_{+}$ is a parameter to be adjusted for making trade-off between the image safety and the distance safety.

First, we claim that the above $h(t)$ is a VS-CBF, since we can always find a sufficiently large control input $\mathbf{u}(t)$ to satisfy the condition (36). Then we show that the VS-CBF ensures the safety of the end-effector in the workspace by the following result.

Proposition 1: The zero-superlevel set $\mathcal{C}$ defined by (41) in (31) satisfies

$$\mathcal{C}(t) \subseteq \mathcal{X}_{S}. \quad (42)$$

With the above results, we have the following corollary.

Corollary 1: Consider the system (3) driven by a controller $\mathbf{u}(t)$ satisfying (36) defined by a VS-CBF in (41). Under Assumption 1, the end-effector can always keep in the safe states, i.e.,

$$\boldsymbol{\xi}(t) \in \mathcal{X}_{S}, \ \forall t \geq t_0. \quad (43)$$

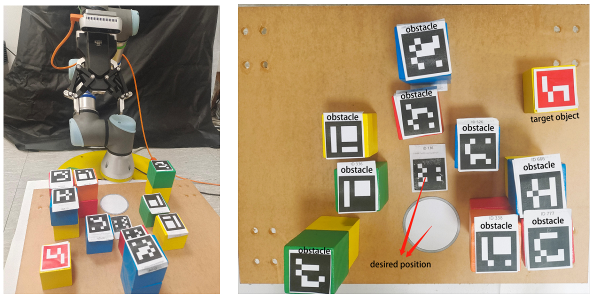

Fig. 1. Experiment Setting

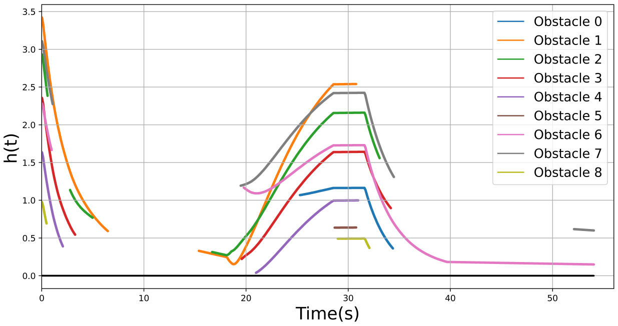

Fig. 2. The Evolution of VS-CBF $h_i(t)$

B. QP-based Safe Visual Servoing Control

With the above results, we can obtain a safe kinematic controller that achieves the specific objective and avoids the collision by solving the following quadratic programming (QP) optimization problem:

$$\min_{\mathbf{u}(t)\in\mathbb{R}^n} \quad \|\mathbf{u}(t)-\hat{\mathbf{u}}(t)\|, \tag{44a}$$

$$\text{subject to Condition (36)}, \tag{44b}$$

where $\hat{\mathbf{u}}(t)$ is computed according to (27) and the VS-CBF $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ is defined in (41).

Now, we summarize the main result of this paper as the following theorem whose correctness has been supported by all the above results.

Theorem 3: Under Assumption 1 and 2, the system (1) driven by the controller obtained in (44) with the nominal controller in (27) and the VS-CBF (41) satisfies

$$\lim_{t\to\infty} \Delta\mathbf{y}(t) = 0, \ \boldsymbol{\xi}(t) \in \mathcal{X}_S, \ \forall t \geq t_0. \tag{45}$$

Remark 4: In this paper, we use the simple VS-CBF in the form of (41) by over-approximating each obstacle as a circle. In fact, there are still many different approaches in the literature for designing a new $h_{\mathbf{y}}(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi})$ to achieve the collision avoidance in the image space, which has been already reduced to a standard CBF design problem.

Remark 5: Compared to the standard CBF, the VS-CBF we propose in Definition 4 does not depend on the coordinates of both the feature point and the obstacles in the workspace. Moreover, the condition (36) is also updated only based on the information sensed by the uncalibrated camera. Therefore, the proposed CBF-based safe vision servoing control can be implemented in a fully vision-based manner.

Remark 6: In the design of condition (36), we utilize the Assumption 1 to find a fixed bound on the estimation error, i.e., $\|\hat{\boldsymbol{\theta}}_{\mathcal{O}}(t) - \boldsymbol{\theta}_{\mathcal{O}}\| \leq \|\overline{\boldsymbol{\theta}}_{\mathcal{O}} - \underline{\boldsymbol{\theta}}_{\mathcal{O}}\|$. In fact, we can also build a time-varying adaptive bound for the estimation error by using the error $\mathbf{e}_{\mathcal{O}}(t_j; t)$ and the constant $\mathbf{W}_{\mathcal{O}}(t_j)$ satisfying (24), which may reduce more conservativeness of the constraint. We leave this as a direction for the future work.

V. Experiments

In this section, we present the implementation of the proposed CBF-based safe visual servoing control on a UR3 manipulator with an Eye-in-Hand RealSense D435i camera. The experimental setting is shown in Figure 1. The manipulator is tasked with grasping an object and placing it to a desired position. The position of the target object and desired position are denoted as two feature points. Furthermore, there are multiple obstacles in the workspace. For the better recognition of the camera, we use the QR codes to denote the positions of all the obstacles and the feature points. For the nominal kinematic controller (27), we set the desired pixels for the grasping and the placing as $\mathbf{y}_{d1} = [461, 460]^{\intercal}$ and $\mathbf{y}_{d2} = [359, 258]^{\intercal}$, respectively, and design the parameters as $\mathbf{B} = \text{Diag}(5\times10^{-7}, 5\times10^{-7})$, $\mathbf{K} = \text{Diag}(0.004, 0.004)$, $\boldsymbol{\Gamma} = 18000\mathbf{I}_{39\times39}$. Also, for the VS-CBF (41), we set $\lambda = 1\times10^{-5}$ and $d_s = 0.08$ while $\mathbf{y}_{\mathcal{O}}(t)$, $r_{\mathcal{O}}$ are detected in real time on the image and ${}^{c}z^{i}_{\mathcal{O}}(t)$ is obtained in real time by the camera. To implement the VS-CBF (41) for the multiple obstacles, it is sufficient to consider the multiple VS-CBF, i.e.,

$$h_i(t) = \lambda\left((\mathbf{y}^{i}_{\mathcal{O}}(t) - \mathbf{y}_{\xi})^2 - r^2_{\mathcal{O}}\right) + ({}^{c}z^{i}_{\mathcal{O}}(t) - d_s), \tag{46}$$

where $y^{i}_{\mathcal{O}}(t)$ and ${}^{c}z^{i}_{\mathcal{O}}(t)$ denote the center and the depth of the i-th obstacle sensed by the camera.

The evolution of each $h_i(t)$ is given in Figure 2. Note that $h_i(t)$ can be computed only when the obstacles are sensed by the camera. We observe that the VS-CBF keeps $h_i(t) \geq 0$ at all $t \geq t_0$, which effectively ensures the safety of the end-effector in the workspace. The video of the experiments is available at `https://youtu.be/PvpHBNnDDbE`.

VI. Conclusion

In this paper, we solve the safe visual servoing control problem of robotic manipulators with uncalibrated camera by designing a vision-based kinematic controller and proposing the VS-CBF based on the image and depth data sensed by the camera, which provides the provable safety guarantee for the end-effector of the manipulators. We validate the effectiveness of the proposed approach by the experiments conducted on UR3 robots. For the future work, we aim to improve the definition of VS-CBF for manipulators by relaxing the assumption of the bounded camera parameters based on a more detailed analysis of the adaptive estimation error. A learning-based CBF construction approach for manipulators based on the image and depth data will also be under investigation in the future.

References

[1] Hossein Abdi, Golnaz Raja, and Reza Ghabcheloo. Safe control using vision-based control barrier function (V-CBF). In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 782–788, 2023.

[2] Devansh R Agrawal and Dimitra Panagou. Safe and robust observer-controller synthesis using control barrier functions. *IEEE Control Systems Letters*, 7:127–132, 2022.

[3] Devansh Ramgopal Agrawal, Ruichang Chen, and Dimitra Panagou. gatekeeper: Online safety verification and control for nonlinear systems in dynamic environments. *IEEE Transactions on Robotics*, 2024.

[4] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference (ECC)*, pages 3420–3431, 2019.

[5] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[6] Mitchell Black, Georgios Fainekos, Bardh Hoxha, Hideki Okamoto, and Danil Prokhorov. CBFkit: A control barrier function toolbox for robotics applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12428–12434, 2024.

[7] Max H Cohen, Tamas G Molnar, and Aaron D Ames. Safety-critical control for autonomous systems: Control barrier functions via reduced-order models. *Annual Reviews in Control*, 57:100947, 2024.

[8] Wenceslao Shaw Cortez and Dimos V Dimarogonas. Safe-by-design control for euler–lagrange systems. *Automatica*, 146:110620, 2022.

[9] Federica Ferraguti, Chiara Talignani Landi, Andrew Singletary, Hsien-Chung Lin, Aaron Ames, Cristian Secchi, and Marcello Bonfe. Safety and efficiency in robotics: The control barrier functions approach. *IEEE Robotics & Automation Magazine*, 29(3):139–151, 2022.

[10] Riku Funada, María Santos, Ryuichi Maniwa, Junya Yamauchi, Masayuki Fujita, Mitsuji Sampei, and Magnus Egerstedt. Distributed coverage hole prevention for visual environmental monitoring with quadcopters via nonsmooth control barrier functions. *IEEE Transactions on Robotics*, 40:1546–1565, 2023.

[11] Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8352–8358, 2021.

[12] Lijun Han, Hesheng Wang, Zhe Liu, Weidong Chen, and Xiufeng Zhang. Vision-based cutting control of deformable objects with surface tracking. *IEEE/ASME Transactions on Mechatronics*, 26(4):2016–2026, 2020.

[13] Lijun Han, Yuyou Zhang, and Hesheng Wang. Vision-based contact point selection for the fully non-fixed contact manipulation of deformable objects. *IEEE Robotics and Automation Letters*, 7(2):4368–4375, 2022.

[14] Hsin-Ai Hung, Hao-Huan Hsu, and Teng-Hu Cheng. Image-based multi-uav tracking system in a cluttered environment. *IEEE Transactions on Control of Network Systems*, 9(4):1863–1874, 2022.

[15] Shaghayegh Keyumarsi, Made Widhi Surya Atman, and Azwirman Gusrialdi. Lidar-based online control barrier function synthesis for safe navigation in unknown environments. *IEEE Robotics and Automation Letters*, 9(2):1043–1050, 2023.

[16] Hassan K. Khalil. *Nonlinear Systems*. Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 2002.

[17] Danica Kragic, Henrik I Christensen, et al. Survey on visual servoing for manipulation. *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, 15:2002, 2002.

[18] Vince Kurtz, Patrick M Wensing, and Hai Lin. Control barrier functions for singularity avoidance in passivity-based manipulator control. In *IEEE Conference on Decision and Control (CDC)*, pages 6125–6130, 2021.

[19] Leslie Lamport. Proving the correctness of multiprocess programs. *IEEE transactions on software engineering*, (2):125–143, 1977.

[20] Jiacheng Li, Qingchen Liu, Wanxin Jin, Jiahu Qin, and Sandra Hirche. Robust safe learning and control in an unknown environment: An uncertainty-separated control barrier function approach. *IEEE Robotics and Automation Letters*, 8(10):6539–6546, 2023.

[21] Lars Lindemann, Alexander Robey, Lejun Jiang, Satyajeet Das, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *IEEE Open Journal of Control Systems*, 2024.

[22] Simin Liu, Changliu Liu, and John Dolan. Safe control under input limits with neural control barrier functions. In *Conference on Robot Learning*, pages 1970–1980. PMLR, 2023.

[23] Yun-Hui Liu, Hesheng Wang, Chengyou Wang, and Kin Kwan Lam. Uncalibrated visual servoing of robots using a depth-independent interaction matrix. *IEEE Transactions on Robotics*, 22(4):804–817, 2006.

[24] Zhe Liu, Kai Li, Tian Hao, and Hesheng Wang. Visual servoing of rigid-link flexible-joint manipulators in the presence of unknown camera parameters and boundary output. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(8):5096–5105, 2023.

[25] Tamas G Molnar and Aaron D Ames. Safety-critical control with bounded inputs via reduced order models. In *American Control Conference (ACC)*, pages 1414–1421, 2023.

[26] Tamas G Molnar, Ryan K Cosner, Andrew W Singletary, Wyatt Ubellacker, and Aaron D Ames. Model-free safety-critical control for robotic systems. *IEEE Robotics and Rutomation Letters*, 7(2):944–951, 2021.

[27] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *IEEE Conference on Decision and Control (CDC)*, pages 3717–3724, 2020.

[28] Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio A Vela. Synthesis of control barrier functions using a supervised machine learning approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7139–7145, 2020.

[29] Manan Tayal, Rajpal Singh, Jishnu Keshavan, and Shishir Kolathaya. Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach. In *American Control Conference (ACC)*, pages 3722–3727, 2024.

[30] Hesheng Wang, Dejun Guo, Hao Xu, Weidong Chen, Tao Liu, and Kam K Leang. Eye-in-hand tracking control of a free-floating space manipulator. *IEEE Transactions on Aerospace and Electronic Systems*, 53(4):1855–1865, 2017.

[31] Hesheng Wang, Yun-Hui Liu, and Dongxiang Zhou. Dynamic visual tracking for manipulators using an uncalibrated fixed camera. *IEEE Transactions on Robotics*, 23(3):610–617, 2007.

[32] Hesheng Wang, Yun-Hui Liu, and Dongxiang Zhou. Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera. *IEEE Transactions on Robotics*, 24(4):843–857, 2008.

[33] Hesheng Wang, Bohan Yang, Yuting Liu, Weidong Chen, Xinwu Liang, and Rolf Pfeifer. Visual servoing of soft robot manipulator in constrained environments with an adaptive controller. *IEEE/ASME Transactions on Mechatronics*, 22(1):41–50, 2016.

[34] Wei Xiao and Calin Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2021.

[35] Wei Xiao, Calin Belta, and Christos G Cassandras. Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67(5):2267–2281, 2021.

[36] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023.

[37] Fan Xu, Xiangjun Kang, and Hesheng Wang. Hybrid visual servoing control of a soft robot with compliant obstacle avoidance. *IEEE/ASME Transactions on Mechatronics*, 2024.

[38] Fan Xu, Hesheng Wang, Zhe Liu, and Weidong Chen. Adaptive visual servoing for an underwater soft robot considering refraction effects. *IEEE Transactions on Industrial Electronics*, 67(12):10575–10586, 2019.

[39] Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.

[40] Dingran Yuan, Xinyi Yu, Shaoyuan Li, and Xiang Yin. Safe-by-construction autonomous vehicle overtaking using control barrier functions and model predictive control. *International Journal of Systems Science*, 55(7):1283–1303, 2024.

[41] Songyuan Zhang, Oswin So, Kunal Garg, and Chuchu Fan. GCBF+: A neural graph control barrier function framework for distributed safe multi-agent control. *IEEE Transactions on Robotics*, 2025.

[42] Dongliang Zheng, Hesheng Wang, Jingchuan Wang, Xiufeng Zhang, and Weidong Chen. Toward visibility guaranteed visual servoing control of quadrotor uavs. *IEEE/ASME Transactions on Mechatronics*, 24(3):1087–1095, 2019.

Appendix I
Proof of Theorem 1

Consider a Lyapunov function

$$V(t) = \frac{1}{2}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}\Delta\mathbf{y}(t) + \frac{1}{2}\Delta\boldsymbol{\theta}_p^{\mathrm{T}}(t)\boldsymbol{\Gamma}\Delta\boldsymbol{\theta}_p(t) \tag{47}$$

Taking the time derivative along the trajectory of system consisting of (7), (9) and (27) yields (50).

Since $\dot{V}(t) = 0$ implies $\Delta\mathbf{y}(t) = 0$ and $\mathbf{e}_p(t_j; t) = 0$ for each $j = 1, \ldots, k$, by LaSalle's Invariance Principle [16], we know that

$$\lim_{t\to\infty} \Delta\mathbf{y}(t) = 0, \quad \lim_{t\to\infty} \mathbf{e}_p(t_j; t) = 0 \tag{48}$$

Furthermore, based on (24) and the result of [32, Propositions 3 & 4], we know that, if $k \geq 20$, then we have

$$\mathbf{e}_p(t_j; t) = 0 \ \Rightarrow\ \hat{\boldsymbol{\theta}}_p(t) = l\boldsymbol{\theta}_p \tag{49}$$

which implies $\lim\limits_{t\to\infty} \hat{\boldsymbol{\theta}}_p(t) = l\boldsymbol{\theta}_p$. The proof is thus completed.

Appendix II
Proof of Theorem 2

Taking the time derivative of $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ in (29) along the trajectories of (7) and (9) yields (52). Since Assumption 1 is satisfied and the initial condition for the estimation law (34) is set as $\underline{\boldsymbol{\theta}}_{\mathcal{O}} \leq \hat{\boldsymbol{\theta}}_{\mathcal{O}}(t_0) \leq \overline{\boldsymbol{\theta}}_{\mathcal{O}}$, by the convergence of the estimation, we have

$$\|\Delta\boldsymbol{\theta}_{\mathcal{O}}(t)\| = \|\hat{\boldsymbol{\theta}}_{\mathcal{O}}(t) - \boldsymbol{\theta}_{\mathcal{O}}\| \leq \|\overline{\boldsymbol{\theta}}_{\mathcal{O}} - \underline{\boldsymbol{\theta}}_{\mathcal{O}}\|. \tag{51}$$

Under the condition (36), we know that, there exists $\mathbf{u}(t) \in \mathbb{R}^n$ such that (53) holds. By the comparison lemma [16], we know that $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) \geq 0$ for all $t \geq t_0$, which means both $\mathcal{S}(t)$ and $\mathcal{C}(t)$ defined by $h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))$ is forward invariant. The proof is thus completed.

Appendix III
Proof of Proposition 1

We prove this result by contradiction. Suppose $\mathcal{C}(t) \nsubseteq \mathcal{X}_S$, which means that there exists $\boldsymbol{\xi}'(t) \in \mathcal{C}(t)$ such that

$$(\mathbf{y}_{\mathcal{O}}(t) - \mathbf{y}_{\xi'})^2 - r_{\mathcal{O}}^2 + \lambda({}^{c}z_{\mathcal{O}}(t) - d_s) \geq 0 \tag{54}$$

but $\boldsymbol{\xi}'(t) \notin \mathcal{X}_S$. Since $\boldsymbol{\xi}'(t) \notin \mathcal{X}_S$, we know that, the end-effector collides with the obstacle at time t. Then we have $(\mathbf{y}_{\mathcal{O}}(t) - \mathbf{y}_{\xi'})^2 \leq r_{\mathcal{O}}^2$ and ${}^{c}z_{\mathcal{O}}(t) - d_s < 0$, which contradicts with (54). The proof is thus completed.

$$\begin{aligned}
\dot{V}(t) &= -\frac{1}{{}^{c}z(t)}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}(\mathbf{A} - \hat{\mathbf{A}})\dot{\mathbf{q}}(t) - \frac{1}{{}^{c}z(t)}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{B}\Delta\mathbf{y}(t) - \frac{1}{{}^{c}z(t)}\Delta\boldsymbol{\theta}_p^{\mathrm{T}}(t)\mathbf{Y}_p^{\mathrm{T}}\mathbf{B}\Delta\mathbf{y}(t) - \Delta\boldsymbol{\theta}_p^{\mathrm{T}}(t)\sum_{j=1}^{k}\mathbf{W}_p^{\mathrm{T}}\mathbf{K}\mathbf{e}_p(t_j; t) \\
&\overset{(26)}{=} \frac{1}{{}^{c}z(t)}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}\mathbf{Y}_p\Delta\boldsymbol{\theta}_p(t) - \frac{1}{{}^{c}z(t)}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\Delta\mathbf{y}(t) - \frac{1}{{}^{c}z(t)}\Delta\boldsymbol{\theta}_p^{\mathrm{T}}(t)\mathbf{Y}_p^{\mathrm{T}}\mathbf{B}\Delta\mathbf{y}(t) - \Delta\boldsymbol{\theta}_p^{\mathrm{T}}(t)\sum_{j=1}^{k}\mathbf{W}_p^{\mathrm{T}}\mathbf{K}\mathbf{e}_p(t_j; t) \\
&\overset{(24)}{=} -\frac{1}{{}^{c}z(t)}\Delta\mathbf{y}^{\mathrm{T}}(t)\mathbf{B}\hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\Delta\mathbf{y}(t) - \sum_{j=1}^{k}\mathbf{e}_p^{\mathrm{T}}(t_j; t)\mathbf{K}\mathbf{e}_p(t_j; t) \leq 0
\end{aligned} \tag{50}$$

$$\begin{aligned}
\dot{h}(t) &= \frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z(t)}\left(\mathbf{A}_{\mathcal{O}}(t) - \hat{\mathbf{A}}_{\mathcal{O}}(t)\right)\mathbf{u}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\left(\mathbf{a}_{\mathcal{O}}^{\mathrm{T}}(t) - \hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t)\right)\mathbf{u}(t) + \left(\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{A}}_{\mathcal{O}}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t)\right)\mathbf{u}(t) \\
&\overset{(35)}{=} -\left(\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z(t)}\mathbf{Y}_{\mathcal{O}}\Delta\boldsymbol{\theta}_{\mathcal{O}}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\mathbf{Y}_z\Delta\boldsymbol{\theta}_{\mathcal{O}}(t)\right) + \left(\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{A}}_{\mathcal{O}}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t)\right)\mathbf{u}(t) \\
&\geq -\left(\frac{1}{{}^{c}z(t)}\left\|\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\right\|\|\mathbf{Y}_{\mathcal{O}}\| + \left\|\frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\right\|\|\mathbf{Y}_z\|\right)\|\Delta\boldsymbol{\theta}_{\mathcal{O}}(t)\| + \left(\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{A}}_{\mathcal{O}}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t)\right)\mathbf{u}(t)
\end{aligned} \tag{52}$$

$$\dot{h}(t) \geq -B(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t)) + \left(\frac{\partial h_{\mathbf{y}}}{\partial\mathbf{y}_{\mathcal{O}}}\frac{1}{{}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{A}}_{\mathcal{O}}(t) + \frac{\partial h_z}{\partial\, {}^{c}z_{\mathcal{O}}(t)}\hat{\mathbf{a}}_{\mathcal{O}}^{\mathrm{T}}(t)\right)\mathbf{u}(t) \geq -\alpha(h(\mathbf{y}_{\mathcal{O}}(t), \mathbf{y}_{\xi}, {}^{c}z_{\mathcal{O}}(t))) \tag{53}$$