2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Robust Partitioned Visual Servoing for Aerial Manipulation Utilizing Controllable-space Image Planning and Adaptive Image Representation

Mohammad Soltanshah[1], Abolfazl Eskandarpour[1], Mehran Mehrandezh[2], and Kamal Gupta[1]

Abstract— **In the pursuit of object retrieval using an aerial manipulator, developing robust visual servoing techniques in the presence of projection and motion model uncertainties is paramount. This paper proposes a novel approach to conducting image-space planning within the controllable-space of the aerial manipulator. Our new strategy resolves the inherent challenge of adhering to a piecewise linear camera trajectory which is infeasible for an aerial manipulator due to the platform's underactuation and presence of secondary tasks for visual servoing. Through this approach, we introduce center of gravity alignment and camera orientation potential fields without relying on specific degrees of freedom from the arm. Moreover, we introduce a new approach that utilizes an image-resolution scaling technique involving an adaptive virtual camera focal length, leading to a numerically well-conditioned image Jacobian. Our proposed framework maintains robustness to the uncertainty in the intrinsic parameters of the camera. We substantiate the efficacy of our methodology through experiments conducted in a realistic physics-based simulation environment.**

I. INTRODUCTION

We are interested in the task of object retrieval using aerial manipulators (AMs)—unmanned aerial vehicles (UAVs) outfitted with arms, grippers, and sensors, well-suited for inspecting hazardous environments and accessing elevated or low-lying areas without human intervention [1]. Assuming the AM has two onboard cameras: one mounted on the UAV (eye-to-hand) and the second mounted on the manipulator wrist (eye-in-hand), it can effectively handle tasks requiring visual guidance and close proximity manipulation (see Figure 1).

The overall object retrieval task could be subdivided into four phases [2]. The very preliminary phase, which we refer to as phase 0, involves the AM navigating from a home position to an outdoor location via GPS until the target is visible. Phase 1 starts when the target is visible, during which the AM uses the onboard eye-to-hand camera to track and guide itself toward the target. Once the AM has reached the proximity of the target, phase 2 starts, when the AM employs eye-to-hand visual servoing [3] to move further toward the target. In the last phase (phase 3), when the target is within the reachable region of the manipulator, the arm servos to a pre-grasp configuration using the eye-in-hand camera, with the base mostly at the hovering state, moving slightly to evade the arm's singularities. This phase concludes with the arm's end-effector being in a pre-grasp pose, prepared to execute the grasping task. Notably, during phases 0, 1, and 2, the arm is maintained in a fixed (retracted) configuration (i.e. a locked-up arm position). In this paper, we focus only on phases 1 and 3, assuming, for example, that [2] could be used for phase 0 and other existing approaches ([3]) could be employed for phase 2. Our integrated framework is detailed in section III-A.

Visual servoing (VS) indeed plays a crucial role here in guiding the AM toward the target; however, it is hindered by the AM's inherent underactuated dynamics and base-pose uncertainties, impacting camera motion and features' trajectories within the image. To address these challenges, image adjustment techniques or virtual-camera methods have been developed, employing perspective projection models on virtual image planes to achieve decoupled and stabilized dynamics for the image features ([4]–[6]), though they introduce modelling complexity and have limited scope for object variability and occlusions. Another approach involves applying dynamic differential flatness to devise aggressive trajectories for AM systems ([7], [8]), with the image features serving as the flat outputs for second-order visual servoing. [8] explored flat spaces in the context of simplified two-dimensional motions in an AM. Nevertheless, extending this study to three-dimensional flat spaces for AMs presents more sophisticated dynamic modelling equations. Furthermore, this approach overlooks the integration of redundancy control schemes.

Partitioned VS, as an alternative strategy, selectively applies VS principles to controllable degrees of freedom (DOFs) of an AM, such as translational and yaw motion for the UAV base and the arm, relegating non-controllable variables, such as roll and pitch, to lower-level control layers ([3], [9]). Moreover, these methods offer a structured framework for integrating redundancy control techniques [10].

Classical VS suffers from convergence issues and instability when the initial and desired configurations are distant for fixed-base manipulator [11]. To mitigate these issues, [12] proposed generating a sequence of image features to track, derived from camera pose space planning; however, since this was mainly addressed within the realm of fully-actuated fixed-base robotic manipulators, underactuation is not a common trait. Additionally, when a UAV is at a considerable distance from the target, a VS method might encounter numerical inaccuracies and instability. This occurs due to the target's increased depth, resulting in some close-

[1]Mohammad Soltanshah, Abolfazl Eskandarpour, and Kamal Gupta are with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada, mohammad_soltanshah@sfu.ca, eskandar@sfu.ca, kamal@sfu.ca

[2]Mehran Mehrandezh is with the Faculty of Engineering and Applied Science, University of Regina, Regina, SK S4S 0A2, Canada, mehran.mehrandezh@uregina.ca

to-zero values for certain components within the image Jacobian matrix, making it numerically ill-conditioned. [13] suggested employing SVD-based regularization to alleviate the strong ill-conditioning in the task Jacobian. Nonetheless, fine-tuning this method presents challenges, particularly at large distances from the target. These are particularly crucial for AMs, given that instability within the visual servoing controller could trigger the platform's physical instability at high frequencies.

In this paper, our first contribution involves conducting image-space planning using the potential field approach. However, we use them within the controllable state space of AM rather than that in the camera pose space [12]. Our new strategy resolves the inherent challenges of i) adhering to a piecewise linear camera trajectory [12], which is infeasible for an AM due to the platform's underactuation and ii) presence of secondary tasks for visual servoing. Furthermore, through this approach, we introduce potential fields for the Center of Gravity (CoG) alignment[1] and camera orientation adjustments without using any specific DOFs of the manipulator, as it is done when using secondary task constraints. This guarantees a precisely controllable CoG alignment, which improves the AM's overall stability when the arm is in motion. Additionally, it facilitates adjustments to the camera viewing angles without having to define secondary tasks in the VS controller.

Our second contribution enhances the pertinent partitioned VS scheme by introducing a novel approach that adaptively changes the image resolution (image-scaling) by regulating the focal length in the pinhole projection model (i.e., the zooming effect), thereby improving the Image Jacobian's numerical condition. This proves to be very effective, especially at large distances between the camera and the target it is servoing towards.

II. PRELIMINARIES

A. Visual Servoing Forward Kinematics

1) Eye-in-hand Forward Kinematics: The structure of the AM, as depicted in Figure 1, comprises a hexacopter, an arm featuring three DOFs, and two onboard cameras. Let $\mathbf{P}$ denote a point in 3D Cartesian space, expressed as ${}^{C_1}\mathbf{P}$ in the eye-in-hand camera frame $\{C_1\}$ and ${}^{I}\mathbf{P}$ in the fixed inertial frame $\{I\}$. Denoting $\mathbf{T}$ as a homogeneous transformation between two consecutive frames, the forward kinematics between ${}^{I}\mathbf{P}$ and ${}^{C_1}\mathbf{P}$ through an AM structure can be expressed as (1).

$$\begin{pmatrix} {}^{I}\mathbf{P} \\ 1 \end{pmatrix} = \mathbf{T}_B^I \ \mathbf{T}_0^B \ \mathbf{T}_1^0 \ \mathbf{T}_2^1 \ \mathbf{T}_E^2 \ \mathbf{T}_{C_1}^E \begin{pmatrix} {}^{C_1}\mathbf{P} \\ 1 \end{pmatrix} \tag{1}$$

We define $\mathbf{X}_{AM}$ as a generalized AM state vector

$$\mathbf{X}_{AM} = (\mathbf{p}^T \boldsymbol{\phi}^T \mathbf{q}^T)^T \tag{2}$$

where $\mathbf{p}$ and $\boldsymbol{\phi}$ are the flying base position and orientation Euler angles vectors in frame $\{I\}$ (located on the target), and $\mathbf{q}$ is a joint angle vector of the attached manipulator.

[1]For the stability of the AM, its CoG should be aligned with the gravity direction passing through the CoG of the UAV.

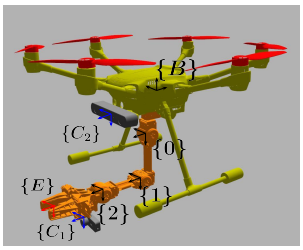


Fig. 1: Aerial manipulator and coordinate frames: body frame $\{B\}$, joint one frame $\{0\}$, joint two frame $\{1\}$, joint three frame $\{2\}$, end-effector frame $\{E\}$, eye-in-hand frame $\{C_1\}$, and eye-to-hand frame $\{C_2\}$

If point $\mathbf{P}$ is located at the origin of frame $\{C_1\}$, the AM Jacobian matrix ($\mathbf{J}_{AM}$) mapping the generalized system velocity vector to the eye-in-hand frame velocity vector expressed in frame $\{I\}$ can be obtained using

$${}^{I}\mathbf{V}_{C_1} = \mathbf{J}_{AM}\dot{\mathbf{X}}_{AM} \tag{3}$$

Let $\boldsymbol{\sigma}_{m_{eih}} = (x, y)^T$ be a vector of eye-in-hand image feature point, and $\mathbf{J}_{I_{eih}}$ be the pinhole eye-in-hand interaction matrix [14] as

$$\mathbf{J}_{I_{eih}} = \begin{bmatrix} \frac{-f}{Z} & 0 & \frac{-x}{Z} & \frac{xy}{f} & -(f+\frac{x^2}{f}) & y \\ 0 & \frac{-f}{Z} & \frac{y}{Z} & (f+\frac{y^2}{f}) & -\frac{xy}{f} & -x \end{bmatrix} \tag{4}$$

where Z is the depth of the target point relative to the eye-in-hand frame, and f is the focal length of the camera. Then, the image feature velocity can be related to the generalized system velocity vector by

$$\dot{\boldsymbol{\sigma}}_{m_{eih}} = \mathbf{J}_{I_{eih}}\mathbf{J}_{AM}\dot{\mathbf{X}}_{AM} \tag{5}$$

While a minimum of three image feature points is imperative to regulate the six DOFs of the eye-in-hand frame, choosing four image feature points helps avoid singularity in the image Jacobian for certain configurations [14].

2) Eye-to-hand Forward Kinematics: Similar to the eye-in-hand configuration, we define $\mathbf{P}$ as a point in 3D Cartesian space, expressed as ${}^{C_2}\mathbf{P}$ in the eye-to-hand frame $\{C_2\}$ and ${}^{I}\mathbf{P}$ in the fixed inertial frame $\{I\}$. The corresponding forward kinematics is written as

$$\begin{pmatrix} {}^{I}\mathbf{P} \\ 1 \end{pmatrix} = \mathbf{T}_B^I \ \mathbf{T}_{C_2}^B \begin{pmatrix} {}^{C_2}\mathbf{P} \\ 1 \end{pmatrix} \tag{6}$$

Also, let $\mathbf{X}_{UAV}$ be a generalized UAV state vector as

$$\mathbf{X}_{UAV} = (\mathbf{p}^T \boldsymbol{\phi}^T)^T \tag{7}$$

Now we can relate the image feature velocity to the generalized UAV velocity vector by

$$\dot{\boldsymbol{\sigma}}_{m_{eth}} = \mathbf{J}_{I_{eth}}\mathbf{J}_{UAV}\dot{\mathbf{X}}_{UAV} \tag{8}$$

where $\boldsymbol{\sigma}_{m_{eth}}$ is a vector of eye-to-hand image feature point, $\mathbf{J}_{I_{eth}}$ is the pinhole eye-to-hand interaction matrix, and $\mathbf{J}_{UAV}$ is the UAV Jacobian matrix.

B. Partitioned Image-based Visual Servoing

1) Eye-in-hand Visual Servoing: Since only four DOFs of a UAV can be commanded, [3] refined equation (5) to accommodate only the UAV controllable DOFs

$$\dot{\boldsymbol{\sigma}}_{m_{eih}} = \mathbf{J}_{m_{eih}} \dot{\boldsymbol{\zeta}} + \bar{\mathbf{J}}_{m_{eih}} \bar{\boldsymbol{\omega}} \tag{9}$$

Here, $\mathbf{J}_{m_{eih}}$ represents the controllable Jacobian of the eye-in-hand system, $\dot{\boldsymbol{\zeta}}$ denotes the rate vector of controllable states for the AM (translational motion of UAV across the fixed reference x, y, z axes, and its yaw motion around frame $\{B\}$), $\bar{\mathbf{J}}_{m_{eih}}$ stands for the non-controllable Jacobian of the eye-in-hand system, and $\bar{\boldsymbol{\omega}} = (\omega_x, \omega_y)^T$ signifies the vector comprising the roll and pitch rates of the UAV.

Considering the desired image feature vector $\boldsymbol{\sigma}^*_{m_{eih}}$ and the image feature error $\tilde{\boldsymbol{\sigma}}_{m_{eih}} = \boldsymbol{\sigma}^*_{m_{eih}} - \boldsymbol{\sigma}_{m_{eih}}$, [9] employed an eye-in-hand redundant partitioned control strategy to nullify $\tilde{\boldsymbol{\sigma}}_{m_{eih}}$ and simultaneously execute a secondary task $\boldsymbol{\sigma}_1$, such as regulating a desired camera orientation, utilizing

$$\dot{\boldsymbol{\zeta}}^* = \mathbf{J}^+_{m_{eih}} \, \boldsymbol{\Lambda}_{m_{eih}} \, \tilde{\boldsymbol{\sigma}}_{m_{eih}} + \left(\mathbf{J}_1 \, \mathbf{N}_{m_{eih}}\right)^+ \boldsymbol{\Lambda}_1 \, \tilde{\boldsymbol{\sigma}}_1 - \bar{\mathbf{J}}_{m|1} \, \bar{\boldsymbol{\omega}} \tag{10}$$

in which $\mathbf{J}^+_{m_{eih}}$ represents the weighted pseudo-inverse of $\mathbf{J}_{m_{eih}}$, $\boldsymbol{\Lambda}_{m_{eih}}$ and $\boldsymbol{\Lambda}_1$ are positive definite gain matrices, $\mathbf{J}_1$ denotes the full rank Jacobian matrix associated with the secondary task, $\mathbf{N}_{m_{eih}}$ is the null space projector of the main task, and $\bar{\mathbf{J}}_{m|1}$ denotes the Jacobian matrix enabling the correction of changes in non-controllable variables given the secondary task.

We enhance the control command (10) by introducing a repulsive potential field aimed at ensuring the target remains within the camera's field of view (FOV), eliminating the need for a separate secondary task. Therefore, we define the distance of the target point from the vertical and horizontal edges of the image as D_x and D_y, respectively. The potential fields [15] described in equations (11) and (12) act to deter the image features from approaching the proximity of the image edges.

$$U_x = \begin{cases} \frac{1}{2}\eta_x(\frac{1}{D_x} - \frac{1}{D^*_x})^2 & D_x \leq D^*_x \\ 0 & D_x > D^*_x \end{cases} \tag{11}$$

$$U_y = \begin{cases} \frac{1}{2}\eta_y(\frac{1}{D_y} - \frac{1}{D^*_y})^2 & D_y \leq D^*_y \\ 0 & D_y > D^*_y \end{cases} \tag{12}$$

D^*_x and D^*_y represent the desired distance of the target point from the vertical and horizontal edges of the image, respectively, while η_x and η_y denote the strength of the repulsive fields. By computing the gradients of (11) and (12) with respect to the image coordinates x and y, we derive the repulsive velocities acting on a feature point as

$$V_{rep_x} = -\alpha_x \, \partial U_x / \partial x \tag{13}$$

$$V_{rep_y} = -\alpha_y \, \partial U_y / \partial y \tag{14}$$

in which α_x and α_y are positive scalars.

We reformulate the control command (10) incorporating the repulsive velocity vector $\mathbf{V}_{rep} = (V_{rep_x}, V_{rep_y})^T$ as

$$\dot{\boldsymbol{\zeta}}^*_{rep} = \dot{\boldsymbol{\zeta}}^* + \mathbf{J}^+_{m_{eih}} \, \boldsymbol{\Lambda}_{rep} \mathbf{V}_{rep} \tag{15}$$

in which $\boldsymbol{\Lambda}_{rep}$ represents a positive definite gain matrix. The control command (15) ensures the VS execution while simultaneously "pushing" the feature point back toward the center of the image when it approaches the margins D^*_x and D^*_y.

2) Eye-to-hand Visual Servoing: Similarly, by following analogous steps, we can express the differential equation and partitioned control command for eye-to-hand visual servoing as equations (16) and (17), respectively:

$$\dot{\boldsymbol{\sigma}}_{m_{eth}} = \mathbf{J}_{m_{eth}} \dot{\boldsymbol{\zeta}}_{UAV} + \bar{\mathbf{J}}_{m_{eth}} \bar{\boldsymbol{\omega}} \tag{16}$$

$$\dot{\boldsymbol{\zeta}}^*_{UAV} = \mathbf{J}^+_{m_{eth}} \, \boldsymbol{\Lambda}_{m_{eth}} \, \tilde{\boldsymbol{\sigma}}_{m_{eth}} - \bar{\mathbf{J}}_{m_{eth}} \, \bar{\boldsymbol{\omega}} \tag{17}$$

In equation (16), $\mathbf{J}_{m_{eth}}$ indicates the controllable Jacobian of the eye-to-hand system, $\dot{\boldsymbol{\zeta}}_{UAV}$ signifies the rate vector of controllable states of the UAV, and $\bar{\mathbf{J}}_{m_{eth}}$ represents the non-controllable Jacobian of the eye-to-hand system. In equation (17), $\mathbf{J}^+_{m_{eth}}$ corresponds to the regular inverse of $\mathbf{J}_{m_{eth}}$, $\boldsymbol{\Lambda}_{m_{eth}}$ represents a positive definite gain matrix, and $\bar{\mathbf{J}}_{m_{eth}}$ stands for the Jacobian matrix enabling correction of changes in non-controllable variables of the UAV.

III. METHODOLOGY

A. Visual Servoing Framework

Figure 2 illustrates the VS framework for phases 1 and 3 of the object retrieval mission. In phase 1 (highlighted within the red dashed-line area), a fiducial marker extraction block initially forms feature vectors from eye-to-hand RGBD data. These feature vectors are subsequently relayed to the adaptive image representation block, which produces adaptive image features and an adaptive virtual focal length (details are explained in the section III-B). These outputs are then forwarded to both the high-level VS controller and a UAV controllable state planner. The subsequent intermediate UAV controllable state is projected onto the image space to calculate the next set of desired intermediate image features.

We have assumed that a low-level AM controller is available that can make the robot follow a given trajectory. Any AM controller from the literature could be used here, but we have used one reported in another work from our Lab [16], where we designed a constrained low-level decoupled controller for an AM. A full description is beyond the scope of the work reported in this paper. However, we give a brief description here. In this controller, the cross-coupled dynamics between the UAV and manipulator are modelled as reaction torque and force exerted on the UAV by the manipulator and vice-versa. The inputs of this controller are desired AM controllable velocities, while its outputs include desired AM thrust, moments, and torques. A cascaded Model Predictive Control (MPC) structure is employed for the UAV's controller, addressing first translational dynamics, followed by rotational dynamics. Translational trajectory tracking is managed with a constrained MPC controller, while rotational dynamics are handled using a tube-based Linear Parameter Varying (LPV)-MPC controller to follow the desired rotational trajectory generated by the translational

dynamics controller while managing the reaction torque imposed on the UAV. Similarly, manipulator dynamics are controlled with the LPV-MPC controller to follow a desired trajectory and account for the effects of the UAV on the manipulator.

Phase 3 employs an AM controllable state planner layer (highlighted within the black dashed-line area in Figure 2) to produce intermediate image features forwarded to the high-level image-based controller. Initially, based on the final desired camera orientation and image features, and accounting for problem geometry and constraints, the final desired CoG and AM controllable states are determined. Subsequently, a potential field approach is employed within the controllable space of AM to derive the subsequent desired intermediate AM controllable state. This state is then projected into image space to obtain the subsequent desired intermediate image features. It is noteworthy that in phase 3, the main task of the high-level image-based controller in our approach is to execute VS, with no secondary task assigned. This design choice ensures that specific DOFs of the arm are not used to realize the task. Please note that camera orientation and CoG alignment are taken into account in the global path planning layer in our framework.

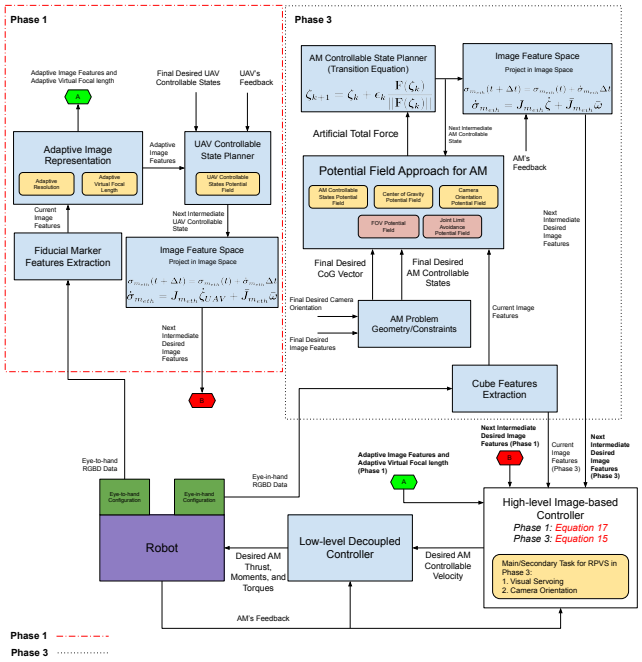


Fig. 2: Aerial manipulator visual servoing framework for phases 1 and 3

B. Phase 1: Adaptive Image Representation and Focal Length

As mentioned earlier, in phase 1, we adapt the image resolution, i.e., the image pixel size P corresponding to the distance d of the target from the AM. Qualitatively, the farther the target is, the lower (or coarser) the image resolution. The precise relationship is described by the equation

$$P(d) = P_{\min} + (P_{\max} - P_{\min})(1 - e^{\frac{-(d-\bar{d})}{k}}) \qquad (18)$$

in which $P_{\max}$ and $P_{\min}$ represent maximum and minimum pixel sizes. d is feature depth, k is the transition rate parameter, and $\bar{d} = \frac{d_{max}+d_{min}}{2}$ is distance threshold, with d_{max} and d_{min} indicating distances corresponding to maximum and minimum pixel size, respectively. Figure 3 shows examples of reconstructing the same RGB image with different resolutions. Noticeably, the coarse resolution depicted in Figure 3-d leads to the merging of four points.

After determining adaptive resolution, we compute the camera's adaptive focal length from terms in $\mathbf{J}_{I_{eth}}$ involving Z in the denominator, $t_1 = \mathbf{J}_{I_{eth_{11}}} = -f_0/Z$, $t_2 = \mathbf{J}_{I_{eth_{13}}} = -x/Z$, and $t_3 = \mathbf{J}_{I_{eth_{23}}} = y/Z$. We introduce CN (equation (19)), a condition number-like metric, offering a basic sensitivity ratio within $\mathbf{J}_{I_{eth}}$.

$$\text{CN} = \frac{\max(|t_1|, |t_2|, |t_3|)}{\min(|t_1|, |t_2|, |t_3|)} \qquad (19)$$

We propose the adaptive virtual focal length formula as

$$f_{\text{a}} = f_0 \left(1 + \alpha \left(\frac{P(d) - P_{\min}}{P_{\max} - P_{\min}}\right)^{\gamma} + \beta \cdot R(x, y) \cdot \text{CN}\right) \qquad (20)$$

in which f_0 is the original focal length of the camera, α modulates the adaptation strength based on pixel size, γ represents the nonlinear exponent constant, and β adjusts the regularization term's influence. $R(x, y)$ modulates regularization based on image coordinates, adjusting for varying ill-conditioning in the Image Jacobian caused by features closer to the image origin.

Noteworthy, when the robot is in close proximity to the target, $P(d)$ approaches $P_{\min}$ and the condition number-like metric, CN, is low, resulting in $f_{\text{a}} \approx f_0$ for fine resolution. Otherwise, equation (20) employs a coarse resolution and higher CN to calculate a greater virtual focal length.

C. Phase 3: Image Space Path Planning for Aerial Visual Servoing

As related to the global image space path planning layer of our framework, one should recall that this is needed to avoid large overshoots in the image feature trajectories when the initial and desired image features are distant. The key idea is to plan a trajectory in image space that is also feasible for the camera to follow. This method [12] ensures the current image features stay near their next desired values while the robustness is enhanced against modelling errors. [12] used the potential field approach to compute a piecewise linear camera trajectory while imposing repulsive potential fields on the camera's FOV and fixed-base arm joint limits. In our framework, shown in Figure 2, we do not need a camera space planner because we plan directly in controllable state space. The global[2] path planning layer has four submodules: computing the controllable state goal configuration, adding the potential field terms to the AM controllable state planner,

[2]Although potential fields can have local minima, in our several simulations, we have not encountered any local minima as has been also reported by [12] for the six-dimensional camera pose problem. They attribute this to the high dimensionality of the underlying space, and in our case, the dimensionality of the space is even higher, with seven-dimensional forces in AM controllable state space.

determining the next intermediate AM state, and projecting the input into the image space to derive the next intermediate image feature vector. We now describe each module in further detail.

(a) orig. pixel size: 2 μm (b) pixel size: 20 μm

(c) pixel size: 25 μm (d) pixel size: 100 μm

Fig. 3: Examples for the reconstruction of the same RGB eye-to-hand image using different image resolutions. The actual scene includes a target with four points. Note that in images (b), (c), and (d), all pixels other than the target features, i.e. white dots, have been zeroed out.

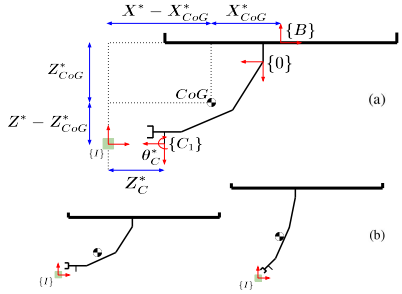


Fig. 4: (a) VS Geometry and constraints for final AM configuration. (b) The left final configuration is computed for inputs of $\theta^*_C = 0$ and $Z^*_C = 0.12m$, while for the right final configuration we input $\theta^*_C = -\pi/3$ and $Z^*_C = 0.12m$.

1) Controllable States Goal Computation: Figure 4-a displays the final configuration for phase 3, computed numerically for five VS geometry and constraints to derive desired AM controllable states using equations (21), assuming zero UAV attitude and position along the Y axis in the $\{I\}$ frame.

$$\begin{aligned} &X^*_{\text{CoG}} = H_1(\mathbf{q}), \quad Z^*_{\text{CoG}} = H_2(\mathbf{q}), \quad \theta^*_C = H_3(\mathbf{q}), \\ &X^* = H_4(Z^*_C, \mathbf{q}), \quad Z^* = H_5(Z^*_C, \mathbf{q}) \end{aligned} \tag{21}$$

H_{1-5} are scalar nonlinear functions (derived using simple straightforward algebraic computations; we omit the details due to space limitations), X^*_{CoG} and Z^*_{CoG} are the final center of gravity coordinates expressed in frame $\{B\}$, θ^*_C is the final camera orientation, and X^* and Z^* are the final UAV position coordinates expressed in frame $\{I\}$. Z^*_C signifies the target's final depth relative to the eye-in-hand frame, computable if the target's geometry is known. Equations in (21) require at least two input variables, typically the final camera orientation and target depth. Figure 4-b illustrates two final configurations with different camera orientations but the same target depth. Note that when the camera orientation is non-zero, the UAV approaches the target more closely, and the manipulator is less extended due to CoG alignment constraint.

2) Attractive Potential Fields: We define three potential fields for the planner's attractors. Initially, a weighted potential field and its corresponding artificial force for AM controllable states are defined by equations (22) and (23).

$$U_{AM} = \frac{1}{2}\, K_{AM}\, \|\mathbf{W}(\mathbf{d}, \bar{\alpha})(\boldsymbol{\zeta} - \boldsymbol{\zeta}_d)\|^2 \tag{22}$$

$$\mathbf{F}_{AM} = -\vec{\nabla}^T_{\boldsymbol{\zeta}} U_{AM} = -K_{AM}\, \mathbf{W}(\mathbf{d}, \bar{\alpha})(\boldsymbol{\zeta} - \boldsymbol{\zeta}_d) \tag{23}$$

K_{AM} represents the strength of AM potential field, and $\boldsymbol{\zeta}_d$ denotes the final AM controllable state vector. $\mathbf{W}(\mathbf{d}, \bar{\alpha}) = \text{diag}((1 - \bar{w})\ \mathbf{I}_{UAV}, \bar{w}\ \mathbf{I}_{ARM})$ is a time-varying diagonal weighting matrix for the UAV and arm motion [9] with

$$\bar{w} = \frac{1 + \bar{\alpha}}{2} + \frac{1 - \bar{\alpha}}{2} \tanh\Big(2\pi \frac{d - \delta_w}{\Delta_w - \delta_w} - \pi\Big) \tag{24}$$

in which $\bar{\alpha}$ indicates a maximum expectation for the UAV involvement in the AM motion, and δ_w and Δ_w are distance thresholds from the target.

Next, we define a CoG potential field and its corresponding artificial force using equations (25) and (26).

$$U_{CoG} = \frac{1}{2}\, K_{CoG}\, \|\mathbf{CoG}(\boldsymbol{\phi}, \mathbf{q}) - \mathbf{CoG}_d\|^2 \tag{25}$$

$$\mathbf{F}_{CoG} = -\mathbf{J}^+_{CoG}\, \big(\vec{\nabla}^T_{\mathbf{q}} U_{CoG} + \vec{\nabla}^T_{\boldsymbol{\phi}} U_{CoG}\big) \cdot \text{Sat}(\mathbf{F}_{AM}) \tag{26}$$

$\mathbf{CoG}$ and $\mathbf{CoG}_d$ represent the current and final AM center of gravity vectors, respectively. K_{CoG} denotes the strength of the CoG potential field, $\mathbf{J}_{CoG}$ represents the CoG task Jacobian, and Sat(.) is a Gaussian function that restricts the influence of U_{CoG} when the U_{AM} value is relatively high.

Finally, we introduce a camera orientation potential field and its corresponding artificial force using

$$U_C(\boldsymbol{\sigma}_C) = \frac{1}{2}\, K_C\, \|\boldsymbol{\sigma}_C - \boldsymbol{\sigma}_{C_d}\|^2 \tag{27}$$

$$\mathbf{F}_C = -\mathbf{J}^+_c\, \vec{\nabla}^T_{\boldsymbol{\sigma}_C} U_C = -K_C\ .\ \mathbf{J}^+_C\ .\ (\boldsymbol{\sigma}_C - \boldsymbol{\sigma}_{C_d})\ .\ \text{Sat}(\mathbf{F}_{AM}) \tag{28}$$

where $\boldsymbol{\sigma}_{C_d}$ is the vector representing the final camera orientation, and $\boldsymbol{\sigma}_C = [\phi_C\ \theta_C\ \psi_C]^T$ denotes the vector for the current camera orientation. K_C signifies the strength of the potential field, and $\mathbf{J}_C$ represents the Jacobian relating the rate of AM controllable states to the rate of camera orientation.

The total attractive artificial force is obtained by summing the contributions from the three attractors, as shown in (29).

$$\mathbf{F}_{att} = \mathbf{F}_{AM} + \mathbf{F}_{CoG} + \mathbf{F}_C \tag{29}$$

3) Repulsive Potential Fields: We incorporate the avoidance of the arm's joint limits and field of view constraints as repulsive fields in the planning approach. To define mechanical constraints for a manipulator, [12] utilized equation (30).

$$U_q(\mathbf{q}) = \prod_{j=1}^{n} (q_j - q_{j_{max}} + l_j)(q_j - q_{j_{min}} + l_j) \tag{30}$$

n is number of joints, l_j is the distance of influence of joint j, q_j and $q_{j_{max}}$ are the current value and maximum allowable value of joint j. The corresponding artificial force is computed by equation (31).

$$\mathbf{F}_q = -\vec{\nabla}^T_{\mathbf{q}} U_q \tag{31}$$

Moreover, [12] described the FOV repulsive potential field as equation (32).

$$U_s(\boldsymbol{\sigma}_{m_{eih}}) = \prod_{j=1}^{m} (u_j - u_{max} + \kappa)(u_j - u_{min} - \kappa)(v_j - v_{max} + \kappa)(v_j - v_{min} - \kappa) \tag{32}$$

m denotes the number of image feature points, and κ represents the distance of influence of the image boundary. The corresponding artificial force is calculated using equation (33).

$$\mathbf{F}_s = -\mathbf{J}^+_{m_{eih}} \vec{\nabla}^T_{\boldsymbol{\sigma}_{m_{eih}}} U_s \tag{33}$$

The total repulsive artificial force is the sum of the two attractors, and the total artificial force is the sum of the attractive force and the repulsive force, as in equation (34).

$$\mathbf{F} = \mathbf{F}_{att} + \mathbf{F}_q + \mathbf{F}_s \tag{34}$$

4) AM Controllable State Planner: After computing the total artificial force, we employ the transition equation (35) to determine the next intermediate AM controllable state.

$$\boldsymbol{\zeta}_{k+1} = \boldsymbol{\zeta}_k + \epsilon_k \frac{\mathbf{F}(\boldsymbol{\zeta}_k)}{||\mathbf{F}(\boldsymbol{\zeta}_k)||} \tag{35}$$

$\boldsymbol{\zeta}_k$ and $\boldsymbol{\zeta}_{k+1}$ represent the current and next AM controllable states, respectively, while ϵ_k denotes the step size that regulates the magnitude of the step in the direction of the normalized vector $\mathbf{F}(\boldsymbol{\zeta}_k)/||\mathbf{F}(\boldsymbol{\zeta}_k)||$.

5) Image Projection Formulation: Once the next intermediate AM controllable state is computed, we use the image projection formulation (36) and (37) to find the next intermediate image features.

$$\dot{\boldsymbol{\sigma}}_{m_{eih}} = \mathbf{J}_{m_{eih}} \dot{\boldsymbol{\zeta}} + \bar{\mathbf{J}}_{m_{eih}} \bar{\boldsymbol{\omega}} \tag{36}$$

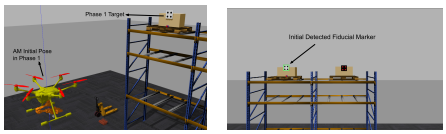
$$\boldsymbol{\sigma}_{m_{eih}}(t + \Delta t) = \boldsymbol{\sigma}_{m_{eih}}(t) + \dot{\boldsymbol{\sigma}}_{m_{eih}} \Delta t \tag{37}$$

IV. RESULTS

We have developed our methodology using the C++ programming language within the ROS framework and Gazebo as the physics-based simulation environment. This section presents the simulation outcomes for phases 1 and 3 of the object retrieval task. Our simulation setup includes models of the Open Manipulator X [17], Typhoon hexacopter [18], and ZED cameras [19]. The robotic arm comprises three active planar revolute joints.

A. Phase 1 Experiment

The effectiveness of phase 1 in our framework is assessed through an experiment positioning the hovering robot 5.1 meters from the target (Figure 5-a), featuring a 30×30 cm^2 fiducial marker with four black points as image features (Figure 5-b).



(a) (b)

Fig. 5: Phase 1 experiment scene: (a) AM initial pose and the target (b) Initial eye-to-hand feature extraction

The gain for the high-level eye-to-hand VS controller is configured as $\boldsymbol{\Lambda}_{m_{eth}} = \Lambda_{m_{eth}} \text{diag}(\mathbf{I}_8)$, where

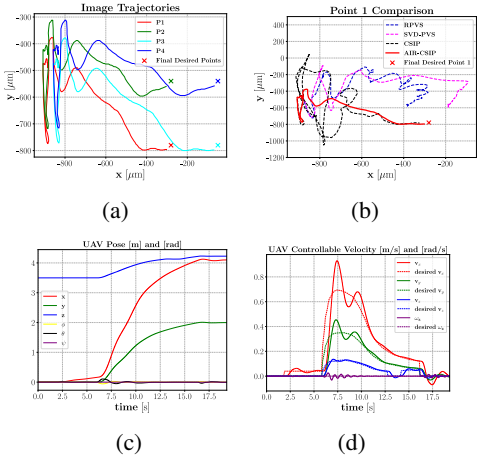
$$\Lambda_{m_{eth}} = \Lambda_{max} \left(\frac{1}{1 + e^{(-k_1(\bar{Z} - k_2))}} \right) + \Lambda_{min} \tag{38}$$

Maximum and minimum control gains are set to $\Lambda_{max} = 7.0$ and $\Lambda_{min} = 1.0$, respectively. $\bar{Z}$ is the average depth of four points in the image, and $k_1 = 2.0$ and $k_2 = 2.5$ are exponential scalars.

In equation (18), we set $P_{\text{max}} = 25\ \mu m$, $P_{\text{min}} = 3\ \mu m$, $d_{\text{max}} = 4\ m$, $d_{\text{min}} = 1\ m$, $f_0 = 2.12\ mm$, and $k = 0.3$. For α in equation (20), one suitable choice is the ratio of the initial depth of the feature to the maximum distance by which we are confident the image Jacobian inverse remains numerically accurate. Hence, we opt for $\alpha = \frac{5.1\ m}{1.2\ m}$. Additionally, we set $\gamma = 0.5$ and $\beta = 0.1$.

Figure 6-a illustrates the image trajectories for the feature points generated by our proposed method, Adaptive Image Representation with Controllable-Space Image Planning (AIR-CSIP). In Figure 6-b, we present a comparison between our two methods AIR-CSIP and Controllable-Space Planning (CSIP), and two baseline methods: Regular Partitioned VS (RPVS) using equation (17), and SVD-based Partitioned VS (SVD-PVS) using (17) with SVD-based inverse of controllable Jacobian. It is evident that when the robot is initially positioned further away from the target (5.1 m) in phase 1, numerical inaccuracies adversely affect the performance of the VS controller, resulting in fluctuating image trajectories. In contrast, AIR-CSIP exhibits smoother image trajectories. Noteworthy, the control gains for RPVS and SVD-PVS are lower compared to our methods, which accounts for the increased fluctuation of CSIP. Figures 6-c and 6-d depict the UAV pose, along with controllable desired and measured velocity vectors for our method, AIR-CSIP. In this phase 1 study, we assume the robot starts from hover and halts at a distance of one meter from the target.

Table I compares partitioned VS methods in phase 1 with respect to final image feature error, completion time, and image trajectory smoothness. AIR-CSIP shows the lowest final image feature error (average error of four points) and fastest completion, outperforming CSIP, RPVS, and SVD-PVS. CSIP faces initial inaccuracies, emphasizing the need for adaptive image representation. We evaluate image trajectory smoothness using a metric which is the average of first, second, and third-order derivatives of trajectory data points, normalized according to the differing completion times of each method. Table I indicates that AIR-CSIP achieves the lowest metric value, suggesting a smoother image trajectory compared to other methods.



(a) (b) (c) (d)

Fig. 6: Phase 1 partitioned visual servoing: (a) Image feature trajectories for our method, AIR-CSIP, (b) Point 1 image feature trajectory comparison between our method, AIR-CSIP with RPVS, SVD-PVS, and our method for CSIP, (c) UAV pose for AIR-CSIP (d) UAV controllable velocity for AIR-CSIP

TABLE I: Comparison between performance of partitioned VS methods

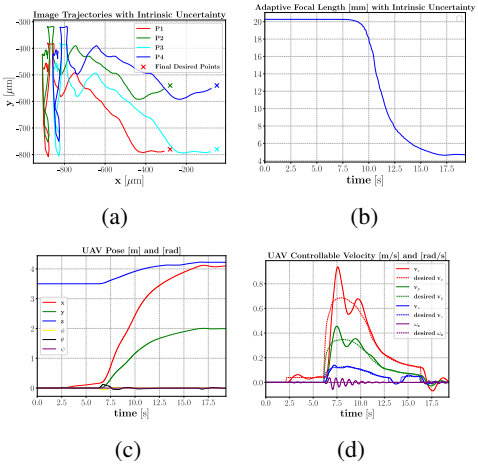
Method	Final Image Feature Error [μm]	Completion Time [s]	Image Trajectory Smoothness Index
RPVS	237.0	20.1	0.049
SVD-PVS	223.7	20.9	0.046
CSIP	44.4	17.9	0.057
AIR-CSIP	27.2	16.2	0.024

B. Phase 1 Experiment with Intrinsic Uncertainty

We now incorporate intrinsic uncertainty into the VS controller. In Figure 7-a, image trajectories for the AIR-CSIP method with 50 percent uncertainty for the original focal length are depicted. It is important to note that in this scenario, RPVS and SVD-PVS failed to complete phase 1. Figure 7-b illustrates the variation in adaptive focal length, with the computed focal length tending toward its original value as the robot approaches the target. Furthermore, Figures 7-c and 7-d shows the UAV pose alongside controllable desired and measured velocity vectors for AIR-CSIP.

C. Phase 3 Experiment - Test Case 1

In phase 3, we assume the AM approaches the target, enabling the arm to extend for a pre-grasp pose (Figure 8-a). The target, a cube with one red face, aids basic image processing for corner extraction, forming image features (Figure 8-b). For test case 1, we strategically positioned initial image features far from the image edges (i.e., closer to the center of the image) to ensure RPVS success with a 40 percent intrinsic uncertainty. Figure 9-a compares image trajectories between our controllable-space image planning (section III-C) and RPVS (equation (15)). It should be noted that our method doesn't define a secondary task in the high-level image-based controller; tasks like camera orientation and CoG alignment are managed only by the upper planning layer (Figure 2, phase 3 dashed-line area). CoG and camera orientation variation using CSIP are shown in Figures 9-c and 9-d, respectively.



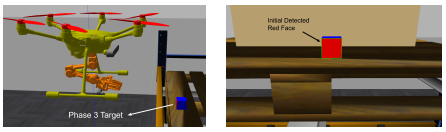
(a) (b) (c) (d)

Fig. 7: Phase 1 partitioned visual servoing with intrinsic uncertainty (a) Image trajectories for our method (AIR-CSIP), (b) Computed adaptive focal length with intrinsic uncertainty, (c) UAV pose, (d) UAV controllable velocity

RPVS may lead to large overshoots in the image due to substantial initial image feature errors which may cause the object to go out of the FOV of the camera. Equation (15) does prevent features from leaving the image but often results in fluctuating VS controller responses, leading to deteriorated performance (Figure 9-a). In contrast, our method (CSIP) generates a smoother trajectory and achieves faster completion (7.6 seconds compared to 15.7 seconds - please refer to the supplemental video). Please note that since our approach (CSIP) takes care of CoG and camera orientation in the global planning stage, for a fair comparison, our RPVS implementation incorporates camera orientation as a secondary task within the high-level image-based controller block depicted in Figure 2.

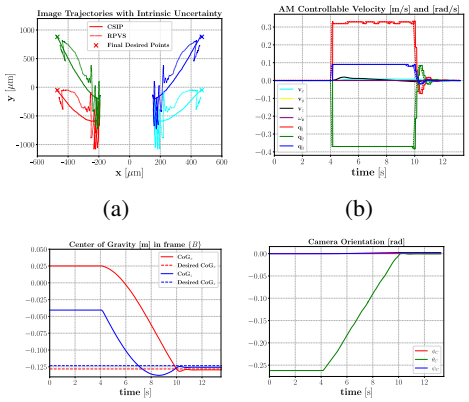
D. Phase 3 Experiment - Test Case 2

In test case 2, the initial target is deliberately positioned closer to the image edge than the image center (Figures 10-a and 10-b), causing RPVS to fail with a 40 percent intrinsic uncertainty. In contrast, our method successfully executes VS. It is worth noting that in test case 2, VS is only accomplished when the UAV also moves; the arm motion alone cannot complete the task. We have selected $\bar{\alpha}$ to be 0.3 in equation (24). $\bar{\alpha}$ is a critical parameter requiring careful adjustment, given its significant impact on the performance of the VS controller. Figure 11-a illustrates the image trajectories for the CSIP method. It is noticeable that at the onset of phase 3, the UAV undergoes a semi-lateral motion adjustment while engaging the arm in VS. This highlights the necessity for the AM to execute non-straight-line camera trajectories to accomplish this VS task. Our method enables the AM to generate controllable-space trajectories customized for its capabilities. In Figure 11-b, the AM desired and measured controllable velocity vectors are illustrated. Compared to test case 1 (Figure 9-b), the average velocities of the arm's joints are lower, emphasizing its reduced contribution to the VS task and highlighting the motion of the UAV.



(a) (b)

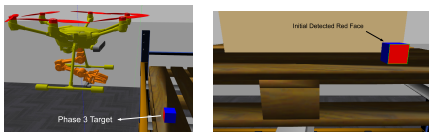
Fig. 8: Phase 3 test case 1 experiment scene: (a) AM initial pose and the target (b) Initial eye-in-hand feature extraction



(a) (b)

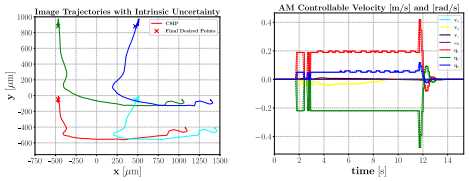
(c) (d)

Fig. 9: Phase 3 test case 1 experiment with intrinsic uncertainty: (a) Image trajectories for CSIP and RPVS, (b) AM controllable velocity for CSIP, (c) CoG variation for CSIP, (d) Camera orientation variation for CSIP



(a) (b)

Fig. 10: Phase 3 test case 2 experiment scene: (a) AM initial pose and the target (b) Initial eye-in-hand feature extraction



(a) (b)

Fig. 11: Phase 3 test case 2 experiment with intrinsic uncertainty: (a) Image trajectories for CSIP (b) AM controllable velocity for CSIP

V. CONCLUSION

This paper introduces novel approaches to object retrieval with an aerial manipulator, focusing on robust partitioned visual servoing amidst numerical inaccuracies and model uncertainties. Our approach adaptively adjusts image resolution and virtual camera focal length to mitigate challenges arising when distances between the camera and the target are large, leading to a numerically well-conditioned image Jacobian. Moreover, by employing controllable image-space planning instead of traditional camera pose space, we introduce center of gravity alignment and camera orientation potential fields in the controllable state space of the AM, thus resolving the inherent challenge of maintaining piecewise linear camera trajectory, which is infeasible for an aerial manipulator. Our experiments conducted within a realistic simulation environment validate the efficacy and robustness of our method when compared to regular partitioned visual servoing techniques for aerial manipulation. Future work will concentrate on extending our work further to the grasping phase of the object retrieval mission as well as implementing it on the real AM hardware, consisting of a hexacopter [18] with an open manipulator X [17] mounted on its underside — the expected challenges are ensuring successful aerial grasping while dealing with uncertainties and payload variations.

REFERENCES

[1] F. Ruggiero *et al.*, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, 2018.

[2] M. Yavari *et al.*, "Interleaved predictive control and planning for an unmanned aerial manipulator with on-the-fly rapid re-planning in unknown environments," *IEEE Trans on Automation Science and Engineering*, 2022.

[3] R. Mebarki *et al.*, "Image-based control for dynamically cross-coupled aerial manipulation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4827–4833.

[4] Y. Chen *et al.*, "Image-based visual servoing of unmanned aerial manipulators for tracking and grasping a moving target," *IEEE Trans on Industrial Informatics*, 2022.

[5] N. Lai *et al.*, "An onboard-eye-to-hand visual servo and task coordination control for aerial manipulator based on a spherical model," *Mechatronics*, vol. 82, p. 102724, 2022.

[6] N. Lai *et al.*, "Image dynamics-based visual servo control for unmanned aerial manipulatorl with a virtual camera," *IEEE/ASME Trans on Mechatronics*, vol. 27, no. 6, pp. 5264–5274, 2022.

[7] J. Thomas *et al.*, "Avian-inspired grasping for quadrotor micro uavs," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55935. American Society of Mechanical Engineers, 2013, p. V06AT07A014.

[8] J. Thomas *et al.*, "Toward image based visual servoing for aerial grasping and perching," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2113–2118.

[9] A. Santamaria-Navarro *et al.*, "Uncalibrated visual servo for unmanned aerial manipulation," *IEEE/ASME Trans on Mechatronics*, vol. 22, no. 4, pp. 1610–1621, 2017.

[10] V. Lippiello *et al.*, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2015.

[11] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 2007, pp. 66–78.

[12] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans on robotics and automation*, vol. 18, no. 4, pp. 534–549, 2002.

[13] A. A. Oliva *et al.*, "Towards dynamic visual servoing for interaction control and moving targets," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 150–156.

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

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[16] A. Eskandarpour *et al.*, "Decoupled dynamic modeling by decomposing the cross-coupled dynamics and tube-based lpv-mpc control scheme for aerial manipulation," *Submitted to IEEE Trans on Aerospace and Electronic Systems*.

[17] Robotis. Openmanipulator-x. [Online]. Available: https://www.robotis.us/openmanipulator-x/

[18] Yuneec. Typhoon hexacopter. [Online]. Available: https://yuneec.online/drones/

[19] Stereolabs. Zed 2 and zed mini cameras. [Online]. Available: https://www.stereolabs.com/products/