Virtual Camera-based Visual Servoing for Rotorcraft using Monocular Camera and Gyroscopic Feedback

Shangkun Zhong, Pakpong Chirarattananon*

Department of Biomedical Engineering, City University of Hong Kong, Hong Kong SAR, China.

Abstract

In this paper, an image-based visual servoing (IBVS) control strategy with a virtual camera frame is proposed for multirotor vehicles. Compared to previous works, the proposed IBVS controller requires minimal sensors in the position loop: a monocular camera and gyroscope. To achieve this, the visual feature projected onto the virtual image plane is associated with the plane's normal instead of relying on an additional attitude estimator and prior knowledge of the plane's inclination. Furthermore, we show that the ratio velocity, when scaled by the image moment, exhibits a similar dynamics to the linear velocity. The finding allows the quantity to be recovered and used for control with a monocular camera without other metric cues. To provide feedback for the controller, an Extended Kalman filter for estimating the ratio velocity, target plane's inclination, and relative rotation between the current and reference camera frame is developed using only monocular vision and gyroscopic measurements. To validate the proposed controller and estimation strategy, both simulation and real-world flight experiments were carried out. The quadrotor smoothly and robustly tracked both dynamic horizontal and static inclined targets, without prior knowledge of the target's inclination. Overall, the proposed regime offers a lightweight and robust alternative IBVS solution for rotorcraft.

Keywords: Micro Aerial Vehicle, visual servoing, robotics

*Corresponding author
Email address: pakpong.c@cityu.edu.hk (Pakpong Chirarattananon*)

1. Introduction

Image-Based Visual Servoing (IBVS) has been successfully deployed with multirotor vehicles [1, 2, 3, 4, 5, 6, 7]. The image-based method aims at directly driving the visual features in the image space to a desired configuration [8], as opposed to first converting the image measurements to the position estimate as carried out in Position-Based Visual Servoing (PBVS) methods. While advances in Visual-Inertial Systems (VINSs) have widely led to reasonably accurate estimates of position and translational velocity, as well as localization [9, 10], these VINSs usually demand substantial computation for the reconstruction of scenes from numerous tracked points. To minimize the number of the tracked features, EKF-based estimators fusing a few image features from the target and IMU data are also employed to provide the relative position estimates for landing [11, 12, 13, 14]. However, this method requires the complex process of converting original image features to the pose and it needs prior knowledge of the structure of the 3D point cloud. Consequently, IBVS methods remain highly attractive for certain applications, taking into account computational efficiency.

Still, the nonlinearity of the camera perspective transformation poses a challenge in designing an IBVS controller for under-actuated multirotor robots. To address this issue, the raw image projection is typically modified to enable the visual feature's kinematics to possess passivity. To date, the control variable of visual features is typically formulated either based on the spherical image projection [15, 2, 3] or image moments accompanied by a virtual camera frame [16, 4]. The use of image moment-based visual features tends to simplify the controller design thanks to the resultant kinematics that is consistent with that of the vehicle's position [4, 17]. To integrate the image moment to the control visual feature, the original visual feature is projected onto an image plane parallel to the ground plane, namely a virtual camera frame, with the help of an external attitude estimator [5, 16, 4, 17].

Applications of IBVS control for aerial vehicles have a practical requirement on state estimation from available onboard sensors, such as a camera and an Inertial Measurement Unit (IMU). To obtain the estimates of the vehicle's velocity in GPS-denied environments, this is typically accomplished with a separate lightweight velocity estimator using optic flow [18, 19, 20] or an IMU sensor in conjunction with the vehicle's model of aerodynamics [21]. Alternatively, a nonlinear velocity observer can be devised in a unified framework within the IBVS controller by means of Lyapunov theory [3, 4]. Nonetheless, these developed velocity observers require an additional attitude estimator to provide the rotation between the camera frame and the inertial frame. Furthermore, the restricted capability of the observer to only estimate the velocity requires the attitude estimator to transform the desired visual features into the inertial frame [3], bringing about an added complication. Otherwise, prior knowledge of the plane's inclination against the ground is needed [4].

To radically simplify the deployment of an IBVS controller on small aerial vehicles by eliminating the need of extra velocity and attitude observers, we take inspiration from applications of Time-To-Contact (TTC), mathematically defined as the ratio of the distance over the velocity, and flow divergence (reciprocal of TTC) for smooth and robust landing [22, 23] or collision avoidance in cluttered environments [24] with rotorcraft. The controllers design based on TTC benefits from the sole dependence on a monocular camera without aided sensory measurements from an accelerometer [19] or vehicle's motor thrust commands [3] as it is no longer necessary to recover the metric scale for estimating the absolute velocity.

This paper offers an alternative IBVS control strategy in conjunction with an EKF-based estimator to position a rotorcraft against a visual target. Compared to existing virtual camera-based methods [5, 16, 4, 17], the proposed framework exhibits highly comparable tracking performance but is the first to incorporate the ratio velocity into the IBVS control law instead of absolute velocity. This offers the benefit of eliminating the dependence on: i) an additional attitude estimator in the position loop, ii) prior knowledge of the target plane's inclination, and iii) metric scale cues for inferring the translational velocity (such as acceleration, or command inputs [3]). To achieve this, the visual feature projected onto the virtual image plane is correlated to the target plane's normal instead of directly aligning the camera plane with the ground. The target plane's normal can be inferred from visual feedback. This dispenses the need for an additional attitude estimator and prior knowledge of plane's inclination. Another beneficial outcome is that the ratio velocity (velocity over the orthogonal distance from the camera to the target) scaled by the image moment exhibits consistent dynamics with the linear velocity, but these variables can be recovered with only monocular camera as found in [25, 23]. The use of ratio velocity, therefore, renders a metric scale cues unnecessary.

To supply the IBVS controller with the feedback of ratio velocity and other quantities required, a robust and computationally efficient estimator is devised. Relying on only a few tracking features, the proposed Extended Kalman Filter (EKF) requires only a gyroscope and a monocular camera to deduce the ratio velocity, target plane's inclination, and the current attitude error. Compared to existing IBVS control frameworks with an onboard observer [3, 4], the proposed control and estimation scheme eliminate the need for both an external attitude estimator in the position control loop and prior knowledge of the target's inclination to generate the desired force commands. To accomplish this, a line segment measurement model is developed to supplement the continuous homography measurement [19] in order to suppress the drift of the estimated target plane's normal when the vehicle is motionless. The proposed control technique arguably shares a similar motivation to [26]. However, the development in [26] was specifically designed for flights in a corridor flight by tracking lines created by the intersection of the side walls and the ceiling/ground whereas the proposed approach exploits point features for the derivation of the control law to track a planar target. This renders the proposed method applicable to more general flight scenarios.

The rest of this paper is structured as follows. Section 2 recalls the dynamic model of multirotor vehicles. Section 3 introduces the image feature and its dynamics. In Section 4, the formulation of the proposed IBVS controller is

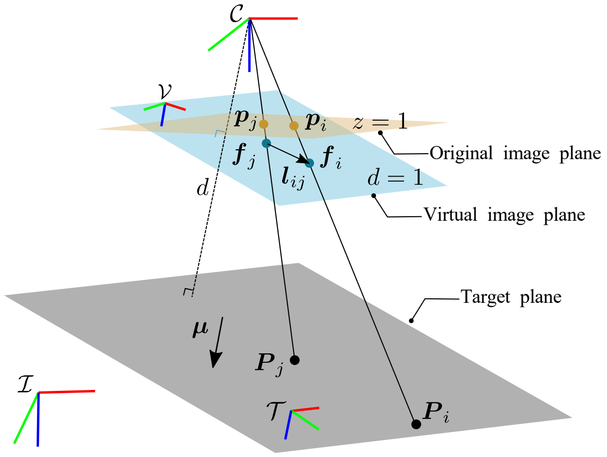


Figure 1: Drawing of the coordinate frames and image projection. The robot is positioned above a planar target. The coordinates $\mathcal{I} = \{O_{\mathcal{I}}, \boldsymbol{x}_{\mathcal{I}}, \boldsymbol{y}_{\mathcal{I}}, \boldsymbol{z}_{\mathcal{I}}\}$ is the inertial frame, $\mathcal{C} = \{O_{\mathcal{CB}}, \boldsymbol{x}_{\mathcal{C}}, \boldsymbol{y}_{\mathcal{C}}, \boldsymbol{z}_{\mathcal{C}}\}$ is the camera frame, $\mathcal{V} = \{O_{\mathcal{V}}, \boldsymbol{x}_{\mathcal{V}}, \boldsymbol{y}_{\mathcal{V}}, \boldsymbol{z}_{\mathcal{V}}\}$ is the virtual camera frame, and $\mathcal{T} = \{O_{\mathcal{T}}, \boldsymbol{x}_{\mathcal{T}}, \boldsymbol{y}_{\mathcal{T}}, \boldsymbol{z}_{\mathcal{T}}\}$ is the target's frame. Red, green and blue lines represents x, y and z axes of the frames. The robot body frame is assumed aligned with the camera frame. A finite number of non-collinear points (shown as $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$) on the target plane with a unit normal $\boldsymbol{\mu}$ are projected onto the image plane of the camera $\mathcal{C}$ at $z = 1$, resulting in a corresponding projection $\boldsymbol{p}_i$. Meanwhile, these points can also be projected onto the virtual image plane that is parallel to the target at a unit distance ($d = 1$) from the camera. A visual feature corresponding to $\boldsymbol{P}_i$, located on the virtual image plane, is defined as $\boldsymbol{f}_i = \boldsymbol{P}_i/d = \boldsymbol{p}_i/\boldsymbol{\mu}^T\boldsymbol{p}_i$. The line segment $\boldsymbol{l}_{ij}$ from the feature $\boldsymbol{f}_j$ to $\boldsymbol{f}_i$ always lies on the virtual image plane and is taken into consideration in the measurement models for the state estimator.

provided, followed by the associated EKF and measurement models in Section 5. In Section 6, simulation experiments were carried out to evaluate the proposed method. In Section 7, extensive flight experiments were performed indoor and outdoor to validate, assess, and benchmark the performance of the proposed method. Lastly, a conclusion is provided. Throughout the manuscript, bold mathematical symbols represent vectors or matrices and regular-weight variables are used to denote scalar quantities.

2. Quadrotor Model

In this section, we briefly recall the quadrotor model. Four coordinate systems are used throughout this paper as shown in Fig. 1: the inertial frame $\mathcal{I} = \{O_{\mathcal{I}}, \boldsymbol{x}_{\mathcal{I}}, \boldsymbol{y}_{\mathcal{I}}, \boldsymbol{z}_{\mathcal{I}}\}$, the robot body frame $\mathcal{B} = \{O_{\mathcal{B}}, \boldsymbol{x}_{\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}}, \boldsymbol{z}_{\mathcal{B}}\}$ attached to the center of mass (CoM), the camera frame $\mathcal{C} = \{O_{\mathcal{C}}, \boldsymbol{x}_{\mathcal{C}}, \boldsymbol{y}_{\mathcal{C}}, \boldsymbol{z}_{\mathcal{C}}\}$ with the camera overseeing the target plane, and the local target frame $\mathcal{T} = \{O_{\mathcal{T}}, \boldsymbol{x}_{\mathcal{T}}, \boldsymbol{y}_{\mathcal{T}}, \boldsymbol{z}_{\mathcal{T}}\}$ located at the center of the fiducial marker. For simplicity, the camera frame is assumed to coincide with the body frame throughout this paper. Let $\boldsymbol{R} \in SO(3)$ denote the rotation matrix from the robot frame $\mathcal{B}$ to the inertial frame and $\boldsymbol{r} \in \mathbb{R}^3$ be the position of the robot in the inertial frame. The rotation $\boldsymbol{R}$ is constructed by three sequential rotations about z-y-x axes with Euler angles ψ, θ and γ denoting the yaw, pitch, and roll, respectively. Both orientation $\boldsymbol{R}$ and position $\boldsymbol{r}$ constitute the 6-degree-of-freedom (DoF) pose describing the transformation between body frame $\mathcal{B}$ and the inertial frame $\mathcal{I}$. Let $\boldsymbol{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ denote the body-centric translational and angular velocities of the robot. The motion of the quadrotor with mass m and moment of inertial $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is dictated by the net force $\boldsymbol{F} \in \mathbb{R}^3$ and torque $\boldsymbol{\Gamma} \in \mathbb{R}^3$ acting on the robot (defined in the body frame) according to

$$\dot{\boldsymbol{r}} = \boldsymbol{R}\boldsymbol{v}, \quad m\dot{\boldsymbol{v}} = -m\left[\boldsymbol{\omega}\right]_\times \boldsymbol{v} + \boldsymbol{F}, \tag{1}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\left[\boldsymbol{\omega}\right]_\times, \quad \boldsymbol{J}\dot{\boldsymbol{\omega}} = -\left[\boldsymbol{\omega}\right]_\times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\Gamma}, \tag{2}$$

where $\left[\boldsymbol{\omega}\right]_\times \in \mathbb{R}^{3\times3}$ represents the skew-symmetric matrix representation of $\boldsymbol{\omega}$. The force $\boldsymbol{F}$, defined in the body frame, is composed of the weight and collective thrust U produced by four motors

$$\boldsymbol{F} = mg\boldsymbol{R}^T\boldsymbol{e}_z - U\boldsymbol{e}_z, \tag{3}$$

with $\boldsymbol{e}_z = [0, 0, 1]^T$ and $g = 9.8$ ms^{-2}. Anticipating non-aggressive flight trajectories, aerodynamic drag is neglected in Eqs. (1) and (2).

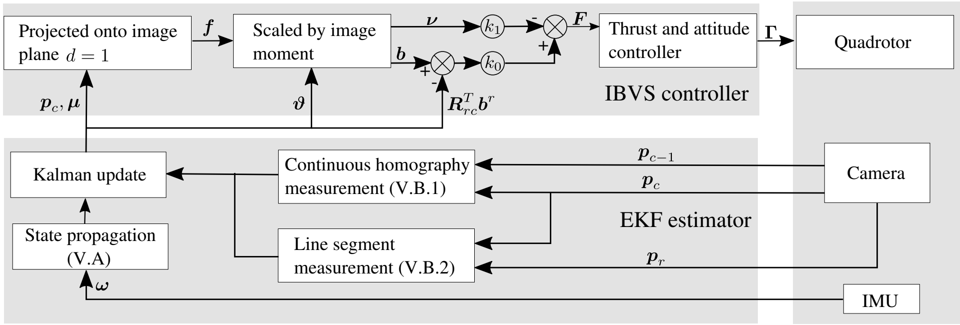


Figure 2: Pipeline of the proposed IBVS controller and estimator.

3. Image Features and Dynamics

The visual feature selection is crucial for the formulation of an IBVS controller since a different choice of visual features leads to distinct system dynamics and therefore the design of the IBVS controller. This section details the development of the proposed visual features for the IBVS controller. The center image moment [27] is leveraged because the incorporation of the center image moment enables the visual feature to behave consistently between the image space and Cartesian space as acknowledged by [2, 16], in contrast to the alternative spherical image feature [3]. However, to address the underactuation of rotorcrafts, the image moment should be calculated with the projection onto an image plane parallel to the target. Consequently, we begin with the image projection onto a virtual image plane with the normal vector identical to that of the target at unit distance to the camera center, notated as the virtual image plane at $d = 1$. Subsequently, the visual feature for IBVS control and its dynamics are derived.

3.1. Image Projection

As illustrated in Fig. 1, a point on the target plane is observed by a camera attached to the quadrotor, resulting in the coordinate $\boldsymbol{P} = [P_x, P_y, P_z]^T$ in the camera frame. The pinhole camera perspectively projects the point onto its image plane at $z = 1$ (this eliminates the need to explicitly include the focal length in the analysis), corresponding to the homogeneous image coordinate $\boldsymbol{p} = \boldsymbol{P}/P_z$. A visual feature $\boldsymbol{f} \in \mathbb{R}^3$ is defined based on the projection of $\boldsymbol{P}$

onto the virtual plane parallel to the target plane at the unit distance from the camera center:

$$\boldsymbol{f} = \frac{\boldsymbol{P}}{d} = \frac{\boldsymbol{P}}{\boldsymbol{P}^T \boldsymbol{\mu}} = \frac{\boldsymbol{p}}{\boldsymbol{p}^T \boldsymbol{\mu}}, \tag{4}$$

with $d = \boldsymbol{P}^T \boldsymbol{\mu}$ denoting the orthogonal distance of the camera center to the target plane and $\boldsymbol{\mu} \in \mathbb{S}^2$ denoting the unit normal vector of the target plane. Since the motions of point $\boldsymbol{P}$ and distance follow

$$\dot{\boldsymbol{P}} = -\left[\boldsymbol{\omega}\right]_\times \boldsymbol{P} - \boldsymbol{v}, \tag{5}$$

$$\dot{d} = -d\boldsymbol{\mu}^T \boldsymbol{\vartheta}, \tag{6}$$

where $\boldsymbol{\vartheta} \in \mathbb{R}^3$ is the ratio velocity defined as $\boldsymbol{\vartheta} = \boldsymbol{v}/d$. The time derivative of $\boldsymbol{f}$ is obtained by differentiating Eq. (4) using Eqs. (5) and (6):

$$\dot{\boldsymbol{f}} = -\left[\boldsymbol{\omega}\right]_\times \boldsymbol{f} - \boldsymbol{\vartheta} + \boldsymbol{f}\boldsymbol{\mu}^T \boldsymbol{\vartheta}. \tag{7}$$

3.2. Visual Features for Control and their Dynamics

As depicted in Fig. 1, a collection of non-collinear points $\boldsymbol{P}_i$ ($i = 1, \ldots, n$, $n \geq 4$) lying on the target plane is projected onto the virtual plane at $d = 1$, resulting in multiple visual features $\boldsymbol{f}_i$. The centroid of these features is $\boldsymbol{q} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{f}_i$ and its time derivative is obtained via with the aid of Eq. (7)

$$\dot{\boldsymbol{q}} = -\left[\boldsymbol{\omega}\right]_\times \boldsymbol{q} - \boldsymbol{\vartheta} + \boldsymbol{q}\boldsymbol{\mu}^T \boldsymbol{\vartheta}. \tag{8}$$

Eq. (8) cannot be directly applied to the IBVS control due to the depth ambiguity. After applying the image moment to the feature centroid $\boldsymbol{q}$, the visual feature for control is obtained as

$$\boldsymbol{b} = \boldsymbol{q}/\beta, \tag{9}$$

where β is the root mean square norm of the difference vector between each feature $\boldsymbol{f}_i$ and their centroid $\boldsymbol{q}$: $\beta^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{f}_i - \boldsymbol{q}\right)^T \left(\boldsymbol{f}_i - \boldsymbol{q}\right)$. Time differentiating β using Eqs. (7) and (8) yields

$$\dot{\beta} = \beta\boldsymbol{\mu}^T \boldsymbol{\vartheta}. \tag{10}$$

Hence, we obtain the time derivative of the control visual feature $\boldsymbol{b}$ by differentiating Eq. (9) and using Eq. (10)

$$\dot{\boldsymbol{b}} = -\left[\boldsymbol{\omega}\right]_{\times} \boldsymbol{b} - \boldsymbol{\nu}, \tag{11}$$

with $\boldsymbol{\nu} = \boldsymbol{\vartheta}/\beta$. Considering the robot's translational dynamics from Eq. (1), the time derivative of $\boldsymbol{\nu}$ can be rewritten as

$$\dot{\boldsymbol{\nu}} = \frac{\dot{\boldsymbol{v}}}{d\beta} = -\left[\boldsymbol{\omega}\right]_{\times} \boldsymbol{\nu} + \frac{1}{mc}\boldsymbol{F}, \tag{12}$$

where the product $c = d\beta$ depends on the relative localization of the features in the world thanks to the definition in Eq. (4) and fact that $\mathrm{d}(d\beta)/\mathrm{d}t = 0$ (according to Eqs. (6) and (10)). Time derivative of $\boldsymbol{\nu}$ possesses the same dynamic behavior to the that of linear velocity (Eq. (1)), but the values of $\boldsymbol{\vartheta}$ and β can be directly inferred from monocular visual cues. Overall, the derivation of the IBVS control in the following section is based on the translational dynamics of the control visual feature in Eq. (11) and (12).

4. IBVS Controller

In this section, we design an IBVS controller through the backstepping method [3, 4] to ensure the stability of the control law. As depicted in Fig. 2, when applied in conjunction with the estimation scheme in the subsequent section, the following formulation does not entail the inertial frame as found in [3, 4, 17] and thereby renders an external attitude estimator dispensable in the outer position control loop.

To address the under-actuation problem of multirotor aerial vehicles, the rotation is generally eliminated from the translational dynamics through rotating the original image plane $z = 1$ to be parallel to the target plane [4] or transforming the desired and current visual feature to the same frame such as the inertial frame [3]. In this paper, the desired visual feature is aligned with the current camera frame [3], $\boldsymbol{\delta}_0 \in \mathbb{R}^3$ is defined as the error of the control feature to be minimized:

$$\boldsymbol{\delta}_0 = \boldsymbol{b} - \boldsymbol{R}_{rc}^{T}\boldsymbol{b}_r, \tag{13}$$

where $\boldsymbol{R}_{rc} \in SO(3)$ is the rotation matrix from the desired camera frame $\mathcal{C}_r$ to the current camera frame $\mathcal{C}_c$ and $\boldsymbol{b}_r$ is the desired centroid, pre-calculated from the visual features at the image plane $z = 1$. These visual features are prepared in advance either by taking an image at the desired pose and performing the feature detection or reprojecting the known 3D points onto the image plane at $z = 1$. Using Eq. (11), we have

$$\dot{\boldsymbol{\delta}}_0 = - [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_0 - \boldsymbol{\nu}, \tag{14}$$

where we have applied the fact that $\frac{d}{dt}\left(\boldsymbol{R}_{rc}^T \boldsymbol{b}_r\right) = - [\boldsymbol{\omega}]_\times \boldsymbol{R}_{rc}^T \boldsymbol{b}_r$. To assure the stability of $\boldsymbol{\delta}_0$, consider the Lyapunov function candidate $V_0 = \frac{1}{2}\boldsymbol{\delta}_0^T \boldsymbol{\delta}_0$ with

$$\dot{V}_0 = \boldsymbol{\delta}_0^T \dot{\boldsymbol{\delta}}_0 = -\boldsymbol{\delta}_0^T \boldsymbol{\nu} - \boldsymbol{\delta}_0^T [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_0 = -\boldsymbol{\delta}_0^T \boldsymbol{\nu}. \tag{15}$$

To guarantee the convergence of V_0, we design the auxiliary input $\boldsymbol{\nu} := k_0 \boldsymbol{\delta}_0$ with k_0 being a positive scalar to be tuned. Consequently, Eq. (15) becomes

$$\dot{V}_0 = -k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 \leq 0, \tag{16}$$

as desired. However, since ν is not the actual input of the system ($\boldsymbol{F}$), to attain $\boldsymbol{\nu} \to k_0 \boldsymbol{\delta}_0$, we introduce the second error term:

$$\boldsymbol{\delta}_1 = \boldsymbol{\delta}_0 - \boldsymbol{\nu}/k_0, \tag{17}$$

or $\boldsymbol{\nu} = k_0 \left(\boldsymbol{\delta}_0 - \boldsymbol{\delta}_1\right)$. Substituting this back into Eq. (14), the time derivative of $\boldsymbol{\delta}_0$ becomes

$$\dot{\boldsymbol{\delta}}_0 = - [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_0 - k_0 \left(\boldsymbol{\delta}_0 - \boldsymbol{\delta}_1\right). \tag{18}$$

The time evolution of $\boldsymbol{\delta}_1 \in \mathbb{R}^3$ is computed from Eqs. (17), (18) and (12),

$$\dot{\boldsymbol{\delta}}_1 = - [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_1 - k_0 \left(\boldsymbol{\delta}_0 - \boldsymbol{\delta}_1\right) - \frac{\boldsymbol{F}}{k_0 mc}, \tag{19}$$

where the system's input $\boldsymbol{F}$ emerges. Next, the revised Lyapunov function including both error terms, $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}_1$, is proposed

$$V_1 = V_0 + \frac{1}{2}\boldsymbol{\delta}_1^T \boldsymbol{\delta}_1 = \frac{1}{2}\boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 + \frac{1}{2}\boldsymbol{\delta}_1^T \boldsymbol{\delta}_1. \tag{20}$$

Using Eqs. (16) and (19), the time derivative of V_1 is

$$\dot{V}_1 = -k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 + k_0 \boldsymbol{\delta}_1^T \boldsymbol{\delta}_1 - \frac{\boldsymbol{\delta}_1^T \boldsymbol{F}}{k_0 mc}. \tag{21}$$

To make $\dot{V}_1$ negative, the chosen control law is

$$\boldsymbol{F} := k_1 k_0 \boldsymbol{\delta}_1, \tag{22}$$

with k_1 denoting a positive scalar to be tuned. Substituting this back into the Eq. (21) yields

$$\dot{V}_1 = -k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0 \right) \boldsymbol{\delta}_1^T \boldsymbol{\delta}_1. \tag{23}$$

Recall from Eq. (12) that c is a constant. With the condition $k_1 > mck_0$, it is sufficient to guarantee that the translational dynamics of the visual feature centroid in Eq. (13) is globally asymptotically stable [28]. To realize the control law in Eq. (22) with a quadcopter, which is not fully actuated, we assume the vehicle is able to reach the desired attitude instantaneously since $\boldsymbol{F}$ is always parallel to the thrust axis (z) of the robot as captured by Eq. (3). For the implementation, the yaw angle is not directly relevant to $\boldsymbol{F}$ and only needs to be stabilized. The desired force $\boldsymbol{F} = [F_x, F_y, F_z]^T$ from control law (Eq. (22)) is converted to roll, pitch and thrust references:

$$\theta^r = \arcsin\left(-\frac{F_x}{mg} \right), \ \gamma^r = \arcsin\left(-\frac{F_y}{mg \cos\theta^r} \right), \tag{24}$$

$$U^r = mg \cos\theta^r \cos\gamma^r - F_z. \tag{25}$$

Subsequently the inner attitude controller is adopted to generate the torque that stabilizes the robot to the reference orientation according to the rotational dynamics in Eq. (2).

As presented in the subsequent section, the required feedbacks of $\boldsymbol{b}$ and $\boldsymbol{\nu}$ for the proposed IBVS method can be recovered using only monocular and gyroscope measurements. This eliminates the need of additional attitude estimator and prior knowledge of plane's inclination to generate the attitude and thrust setpoints compared to existing prior virtual camera-based IBVS methods [5, 16, 4].

5. EKF Estimation Framework

Here, we detail the EKF-based observer for estimating the quantities required by the proposed control law: i) the ratio velocity $\boldsymbol{\vartheta}$, ii) the plane unit normal $\boldsymbol{\mu}$, and iii) the rotation from the reference camera frame to the current camera frame $\boldsymbol{R}_{rc}$. The unified formulation relaxes the dependence on an additional attitude estimator as well as a scale-aided sensor or knowledge, such as an accelerometer and motor thrust used in [4, 17, 3, 5], to generate the desired force command.

In the proposed EKF, the state and covariance propagation is carried out once the gyroscope measurements arrive whereas the main state update step leverages tracked image corners as visual features. We follow the EKF state prediction and update routine detailed in [19].

5.1. State Definition and Prediction

The state vector is composed of the following elements:

$$\boldsymbol{x} := (\boldsymbol{\vartheta}, \boldsymbol{\mu}_s, \boldsymbol{R}_{rc}), \tag{26}$$

where $\boldsymbol{\mu}_s \in SO(3)$ is a representation of the rotation that satisfies $\boldsymbol{\mu} = \boldsymbol{\mu}_s(\boldsymbol{e}_z)$. This implementation resolves the singularity issue and brings about relatively simple differentials [9].

The state prediction begins with the continuous dynamic model of the state $\boldsymbol{x}$:

$$\dot{\boldsymbol{\vartheta}} = \boldsymbol{a}/d + (\boldsymbol{\mu}^T \boldsymbol{\vartheta} \boldsymbol{I}_3 - [\boldsymbol{\omega}]_\times)\boldsymbol{\vartheta} + \boldsymbol{w}_\vartheta, \tag{27}$$

$$\dot{\boldsymbol{\mu}}_s = \boldsymbol{N}(\boldsymbol{\mu}_s)^T \boldsymbol{\omega} + \boldsymbol{w}_\mu, \quad \dot{\boldsymbol{R}}_{rc} = \boldsymbol{R}_{rc} [\boldsymbol{\omega}]_\times, \tag{28}$$

where $\boldsymbol{a}$ is the robot acceleration expressed in the camera frame and $\boldsymbol{I}_3$ is the 3×3 identity. The angular velocity $\boldsymbol{\omega}$ is determined from the readings from the gyroscope measurements $\boldsymbol{\omega}_m$ after subtractions of predetermined biases $\boldsymbol{b}_\omega$ $\boldsymbol{w}_\omega$: $\boldsymbol{\omega} = \boldsymbol{\omega}_m - \boldsymbol{b}_\omega$. The terms $\boldsymbol{w}_i$'s are zero-mean Gaussian white noise. The operator $\boldsymbol{N}(\bullet)^T$ linearly projects a 3D vector into the 2D tangent space of a unit vector in $\mathbb{R}^2$ such that $\boldsymbol{N}(\boldsymbol{\mu}_s) = \left[\boldsymbol{\mu}_s(\boldsymbol{e}_x), \boldsymbol{\mu}_s(\boldsymbol{e}_y)\right]$, where $\boldsymbol{e}_x = \left[1,0,0\right]^T$ and $\boldsymbol{e}_y = \left[0,1,0\right]^T$ so $\boldsymbol{\mu}_s(\boldsymbol{e}_i)$'s become basis vectors of the coordinate system [9]. Neglecting the acceleration term, the dynamics of the ratio velocity in Eq. (27) is approximated as

$$\dot{\boldsymbol{\vartheta}} \approx (\boldsymbol{\mu}^T \boldsymbol{\vartheta} \boldsymbol{I}_3 - [\boldsymbol{\omega}]_\times)\boldsymbol{\vartheta} + \boldsymbol{w}_\vartheta. \tag{29}$$

This means the temporal change in $\boldsymbol{\vartheta}$ attributable to the acceleration is neglected, leaving only the contribution from the linear and angular velocities. The assumption is reasonable for flights without aggressive maneuvers. Moreover, our preliminary results reveal that the deviation of the predicted ratio velocity caused by this approximation can be compensated by the image measurement model thanks to the observability condition of the ratio velocity with a monocular camera [25, 23].

Despite the continuous-time nature of the system, the implementation of EKF in the hardware must be discretized (such as by the forward Euler method). In the discrete-time domain, once a new IMU dataframe arrives at timestamp t_k, the discretized state $\boldsymbol{x}_k$ is propagated from the state $\boldsymbol{x}_{k-1}$ at timestamp t_{k-1}.

5.2. Measurement Models

5.2.1. Continuous Homography Measurement Model

To rectify the prediction drift, the continuous homography equation [20, 19] is applied to link optic flow from two consecutive frames to the current camera state.

At time instance t_{c-1}, a point on the surface is projected through the perspective transformation onto the image plane at $\boldsymbol{p}_{c-1}$. After one camera-based time period δT, the point displaces to a new location $\boldsymbol{p}_c$ on the current (new) image plane according to the motion prescribed by the current state $\boldsymbol{x}_k$. This displacement can be approximately characterized by the continuous homography equation [19, 20]:

$$\bar{\boldsymbol{p}}_c \approx \bar{\boldsymbol{p}}_{c-1} - \delta T[\boldsymbol{I}_2, -\bar{\boldsymbol{p}}_{c-1}]\boldsymbol{H}_k(\boldsymbol{x}_k)\boldsymbol{p}_{c-1} = \boldsymbol{h}_C(\boldsymbol{x}_k), \tag{30}$$

where the operator $\bar{\bullet} : \mathbb{R}^3 \to \mathbb{R}^2$ projects the 3D image coordinate into the inhomogeneous form, $[\boldsymbol{I}_2, -\bar{\boldsymbol{p}}_{c-1}]$ refers to the concatenation of two matrices to form a new matrix, and $\boldsymbol{I}_2$ is a 2×2 identity. The matrix $\boldsymbol{H}_k \in \mathbb{R}^{3\times 3}$ is the continuous homography matrix [19] containing the camera angular velocity, ratio velocity and plane's normal:

$$\boldsymbol{H}_k = -\left[\tilde{\boldsymbol{\omega}}_k\right]_\times - \boldsymbol{\vartheta}_k \boldsymbol{\mu}_k^T. \tag{31}$$

The angular velocity $\tilde{\boldsymbol{\omega}}_k$ is $\boldsymbol{\omega}$ averaged over the camera-based time period δT. The continuous measurement model, Eq. (30), is effective in correcting the inaccurate ratio velocity resulting from the approximation made in Eq. (29). Nevertheless, the drawback of the continuous homographic relationship in Eq. (31) is that the plane's normal becomes ambiguous when the ratio velocity $\boldsymbol{\vartheta}_k$ is near-zero and it is sensitive to the image measurement noise. This issue is alleviated by a supplementary measurement model below.

5.2.2. Line Segment Measurement Model

Due to the reduced observability of the normal vector when the ratio velocity is insufficiently excited, the estimate of the plane normal drifts as soon as the robot becomes static. Besides, the continuous homography model does not provide information about the rotation from the desired camera frame to the current camera frame. Therefore, a line segment measurement model is devised to augment the continuous measurement model.

As illustrated in Fig. 1, we focus on the line segment from a visual feature $\boldsymbol{f}_i$ on the virtual image plane at $d = 1$ to another visual feature $\boldsymbol{f}_j$

$$\boldsymbol{l}_{ij} = \boldsymbol{f}_i - \boldsymbol{f}_j = \frac{\boldsymbol{p}_i}{\boldsymbol{p}_i^T \boldsymbol{\mu}} - \frac{\boldsymbol{p}_j}{\boldsymbol{p}_j^T \boldsymbol{\mu}}. \tag{32}$$

This line segment $\boldsymbol{l}_{ij} \in \mathbb{R}^3$, when observed by the desired $\mathcal{C}_r$ and current camera

$\mathcal{C}_c$ frames are

$$\boldsymbol{l}_{ij}^r = \frac{\boldsymbol{p}_i^r}{\boldsymbol{p}_i^r \cdot \boldsymbol{\mu}^r} - \frac{\boldsymbol{p}_j^r}{\boldsymbol{p}_j^r \cdot \boldsymbol{\mu}^r}, \quad \boldsymbol{l}_{ij}^c = \frac{\boldsymbol{p}_i^c}{\boldsymbol{p}_i^c \cdot \boldsymbol{\mu}^c} - \frac{\boldsymbol{p}_j^c}{\boldsymbol{p}_j^c \cdot \boldsymbol{\mu}^c}, \tag{33}$$

where $\boldsymbol{p}_i^r$ and $\boldsymbol{p}_i^c$ are the image projections of the same points onto the image plane at $z = 1$ in the desired and current frames. The normal $\boldsymbol{\mu}^c = \boldsymbol{\mu}_k$ is the state to be estimated. The normal $\boldsymbol{\mu}^r$ in the desired camera frame is constant and obtained when initializing the estimator using the homography decomposition (the homography initialization requires at least four corners due to the 8-DoF homography matrix) [29].

As derived in Appendix B, the same line segment observed under two camera frames are related by the relative rotation between the frames and a positive scale factor α. That is,

$$\boldsymbol{l}_{ij}^r = \alpha \boldsymbol{R}_{rc} \boldsymbol{l}_{ij}^c = \boldsymbol{h}_L(\boldsymbol{x}_k). \tag{34}$$

Eq. (34) serves as a second measurement model. The scale factor α is to be marginalized out during the update stage using the technique found in [9].

5.3. Measurement Vector and State Update

Suppose n corners are employed to form a set of visual features, both continuous homography and line segment measurement models are applied to construct the measurement vector $\boldsymbol{z}_k$ by stacking vectors $\boldsymbol{h}_C(\boldsymbol{x}_k)$ from Eq. (30) and $\boldsymbol{h}_L(\boldsymbol{x}_k)$ from Eq. (34). The homography model constitute $2n$ elements in $\boldsymbol{z}_k$ as $\bar{\boldsymbol{p}}_c \in \mathbb{R}^2$ (see Eq. (30)). The line segment model produces $n(n-1)/2$ lines, each contribute to three elements in $\boldsymbol{z}_k$. In total, this yields $\boldsymbol{z}_k \in \mathbb{R}^{(3n^2+n)/2}$ as

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{\eta}_k, \tag{35}$$

where $\boldsymbol{\eta}_k$ is the observation noise assumed to be zero-mean Gaussian white noise with covariance $\boldsymbol{\Omega}_k$. Subsequently, the Kalman update is carried out:

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- \boxplus \Delta \boldsymbol{x}_k, \quad \Delta \boldsymbol{x}_k = \boldsymbol{K}_k \left(\boldsymbol{z}_k - \boldsymbol{h}(\boldsymbol{x}_k^-) \right). \tag{36}$$

The boxplus (or boxminus) operator in Eq. (36) behaves as a regular addition (or subtraction) in the Euclidean space, except when it is applied to unit vectors defined on 2-manifolds ($\mathbb{S}^2$). Readers are referred to [9] for the detailed definition of these operators.

Rather than using a regular Kalman gain for $\boldsymbol{K}_k$ in Eq. (36), we use the Gauss-Newton Kalman gain [30, 19], which simplifies the process of the marginalization of α defined in Eq. (34):

$$\boldsymbol{K}_k = \left((\boldsymbol{L}_k^T \boldsymbol{\Sigma}_k^- \boldsymbol{L}_k)^{-1} + \boldsymbol{S}_k^T \boldsymbol{\Omega}_k^{-1} \boldsymbol{S}_k \right)^{-1} \boldsymbol{S}_k^T \boldsymbol{\Omega}_k^{-1}, \tag{37}$$

where matrices $\boldsymbol{L}_k$ and $\boldsymbol{S}_k$ are Jacobians [9]:

$$\boldsymbol{L}_k = \frac{\partial \boldsymbol{x}_k^- \boxplus \Delta \boldsymbol{x}}{\partial \Delta \boldsymbol{x}} \left(\boldsymbol{x}_k^+ \boxminus \boldsymbol{x}_k^- \right), \; \boldsymbol{S}_k = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_k^-)}{\partial \boldsymbol{x}_k^-}. \tag{38}$$

Finally, the state covariance is updated according to

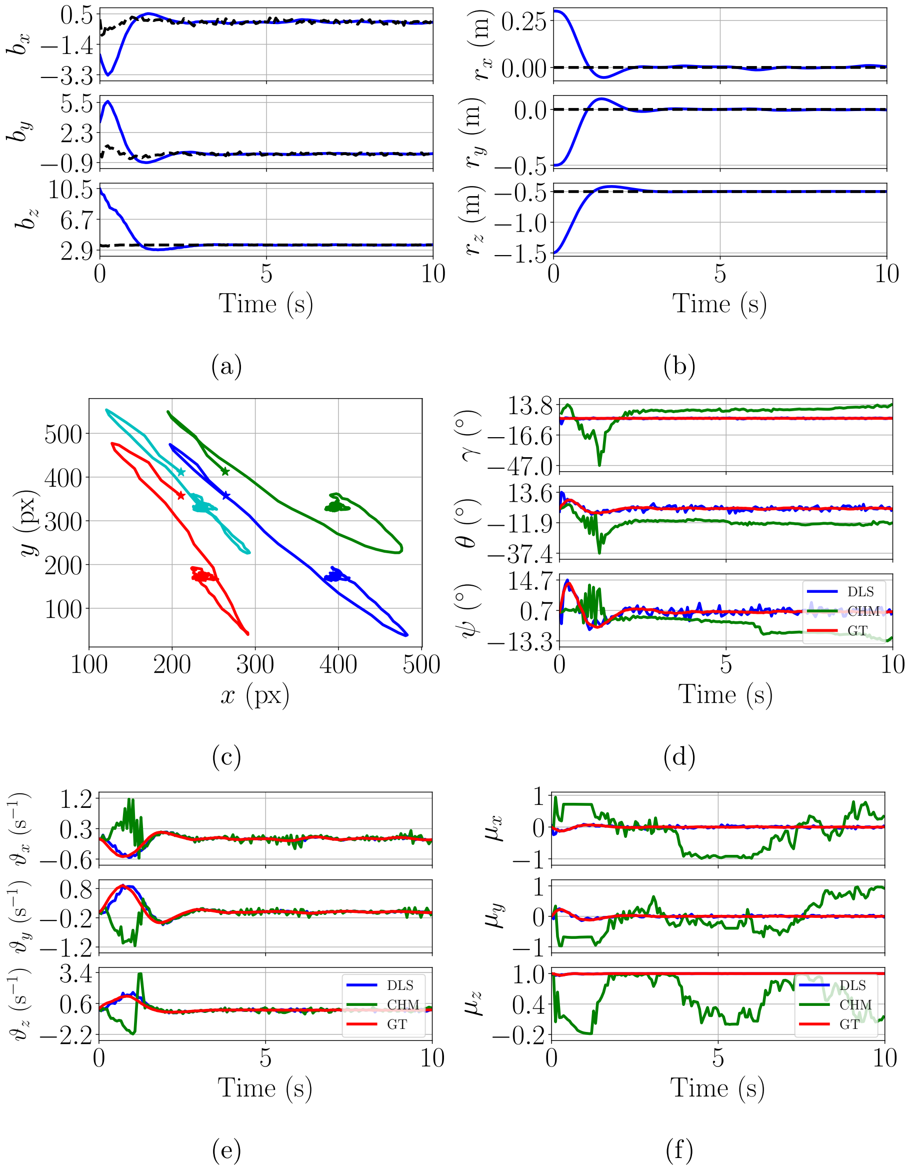
$$\boldsymbol{\Sigma}_k^+ = \boldsymbol{\Sigma}_k^- - \boldsymbol{K}_k \boldsymbol{S}_k \boldsymbol{L}_k^T \boldsymbol{\Sigma}_k^- \boldsymbol{L}_k. \tag{39}$$

Despite some nonlinearity, EKF displays satisfactory performance in practice when it is initialized with fairly accurate estimates of the rotation and plane's normal. This renders further enhancements, such as the iterated EKF scheme used in our previous work [19] unnecessary.

Regular homography model is another option to suppress the drift of the target plane's normal with the quadrotor approaching zero velocity. Nonetheless, that model introduces the unnecessary relative ratio position between two frames to be included in the state vector. A second drawback is that the drift of the normal estimation still occurs for the regular homography when the relative position between the desired and current frame is near-zero.

6. Simulation Results

The simulation was carried out to assess and validate the proposed IBVS controller and EKF estimator. The task is to stabilize the quadrotor with respect to a stationary planar target. Two scenarios were simulated: horizontal and tilted target planes.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: Simulation results of the horizontal target plane: time evolution of (a) the visual feature centroid, (b) position, (c) the feature trajectory in the virtual image plane at $z = 1$ from the initial point (star) to the desired position (solid circle), (d) the estimates of the orientation of the current frame with respect to the desired frame, (e) the estimates of the ratio velocity, and (f) the estimates of the plane's unit normal. For figures (a) and (b), the black dashed lines and blue solid lines represent the setpoints and actual values of the corresponding variables.

Table 1: RMSE from the simulation.

Flight	e_ϑ (s^{-1})		e_μ ($^\circ$)		e_{rot} ($^\circ$)	
	DLS	CHM	DLS	CHM	DLS	CHM
0°	0.1	3.3	2.4	61.2	2.4	16.9
35°	0.1	3.3	1.1	51.4	1.2	35.4

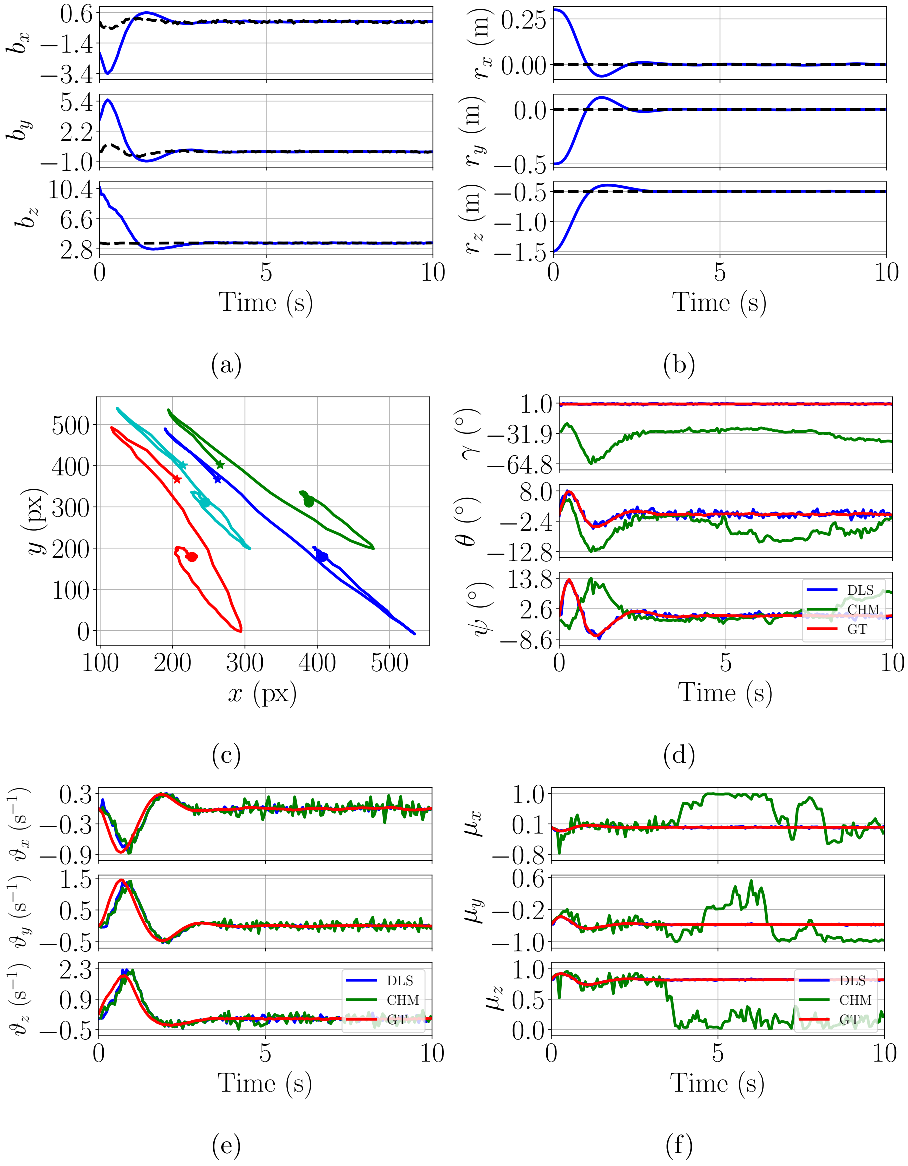
6.1. Simulation Environments

The quadrotor affixed with a downward-facing camera has the mass $m = 0.6$ kg and moment of inertial $\boldsymbol{J} = \mathrm{diag}(9, 9, 12.5)\times 10^{-3}$ kg·m^2. The intrinsics calibration parameters of the camera are identical to the actual camera in the succeeding section, with the focal length of 400 px/m and image size of 640×480 px. The zero-mean Gaussian white noises are simulated for the gyroscope (standard deviation, STD: 1°/s) and camera measurements (STD: 1 px). The high-gain attitude controller in the low-level control loop and gyroscope operates at 500 Hz and the IBVS controller runs at 30 Hz based on the sample rate of the camera. The gains for the IBVS control law are $k_0 = 0.3$ and $k_1 = 0.2$. A squared fiducial maker with length $l = 0.2$ m is employed as the target. Four vertices of the square marker, serving as four visual features, in the target frame $\mathcal{T}$ are at $[-l/2, -l/2, 0]^T$, $[l/2, -l/2, 0]^T$, $[l/2, l/2, 0]^T$ and $[l/2, l/2, 0]^T$.

6.2. Simulation Results using the Horizontal Target Plane

The target was placed at the origin of the inertial frame and the orientation coincides with the inertial frame. The vehicle's initial position was set to $[0.3, -0.5, -1.5]^T$ m. The reference visual centroid $\boldsymbol{b}^r$ and corresponding robot position were $[0, 0, 3.5]^T$ and $[0, 0, -0.5]^T$ m respectively.

As shown in Fig. 3 (a)-(c), the actual control feature, robot position and image features converge to the setpoints after about 3 s. The vehicle remains at the equilibrium point for the rest of time. This qualitatively verifies both the estimation and control framework. Fig. 3 (d)-(f) plot the estimates (DLS) of $\boldsymbol{R}_{rc}$, $\boldsymbol{\vartheta}$, and the plane's normal $\boldsymbol{\mu}$ against the ground truths (GT). The results



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Simulation results of the tilted target plane: time evolution of (a) the visual feature centroid, (b) position, (c) the feature trajectory in the virtual image planeat $z = 1$ from the initial point (star) to the desired position (solid circle), (d) the estimates of the orientation of the current frame with respect to the desired frame, (e) the estimates of ratio velocity, and (f) the estimates of plane's unit normal. For figures (a) and (b), the black dashed lines and blue solid lines represent the setpoints and actual values of the corresponding variables.

further attest the performance of the proposed EKF with two measurements model.

To further inspect the contribution of the line segment model, the set of gyroscope and image measurements obtained from the simulated flight is supplied to the EKF, but without using the line segment model. The estimation results (notated as CHM) are displayed on Fig. 3 (d)-(f). The root mean square errors (RMSE) of the estimated quantities from both cases (DLS and CHM) are listed in Table 1. One can see that the estimates of the plane's normal and $\boldsymbol{R}_{rc}$ from CHM are noticeably inferior whereas the ratio velocity is only marginally affected. As previously mentioned, the continuous homography measurement model does not contain cues about the orientation between the desired and current frame. Meanwhile, the plane's normal cannot be inferred owing to the reduced observability when the vehicle remains stationary.

6.3. Simulation Results using a 35°*-inclined Target*

The target was placed at the origin of the inertial frame and rotated 35° around the x-axis. The setpoint visual centroid becomes $[1, 0.43, 4]^T$ due to plane inclination. The initial conditions and reference position remain identical to those from the horizontal target case.

As shown in Fig. 4 (a)-(f), the results resemble those of the previous case. Fig. 4 (a)-(c) presents that the vehicle's position, control variable $\boldsymbol{b}$ and image features converge to the target configuration in 3 s and subsequently remains stable. According to Fig. 4 (d)-(f), the performance of the estimator without the line segment measurement suffers from notable deviations from the ground-truth, except for estimates of the ratio velocity. The small oscillations of the estimates around the ground-truths are caused by the synthetic white noise. The simulation results collaborate the compatibility of the proposed estimator to a non-horizontal target.

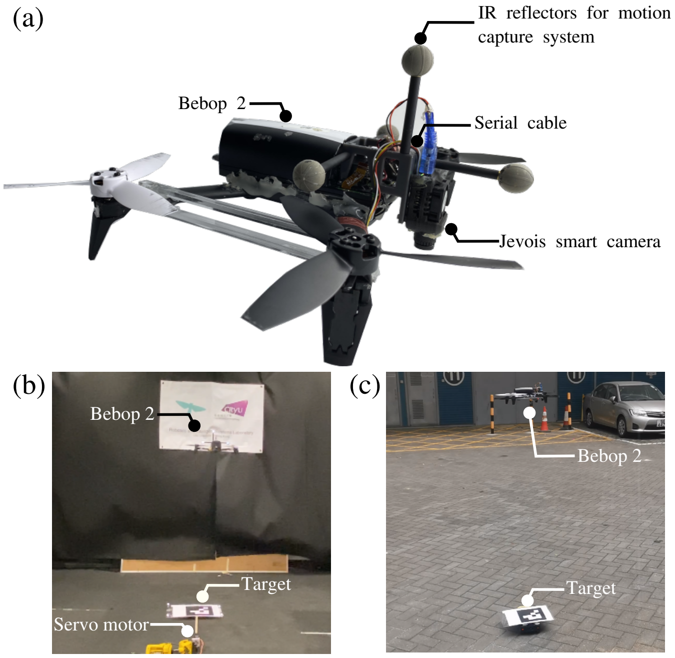


Figure 5: (a) Bebop 2 quadrotor a with downward-facing camera. The Jevois smart camera package consisting of the IMU, monocular camera and processor was used to detect the target and deduce the neccessary feedbacks for the proposed IBVS controller. The data was transmitted from the Jevois camera to the mainboard of Bebop 2 through serial communication. The external motion capture system was used to provide the groundtruth of the estimates. (b) and (c) Snapshots of the indoor and outdoor flight experiments.

7. Real-world Experimental Evaluation

In this section, indoor and outdoor experiments were conducted to assess the performance of the proposed IBVS controller and EKF estimator.

7.1. Experimental Setup

270 As shown in Fig. 5 (a), we employed a commercial quadrotor Parrot Bebop 2 running open-source Paparazzi software[2] as our platform. A motion tracking system (NaturalPoint, OptiTrack) was used to provide the ground truths. A JeVois Smart Machine Vision camera was mounted on the robot to provide vi-

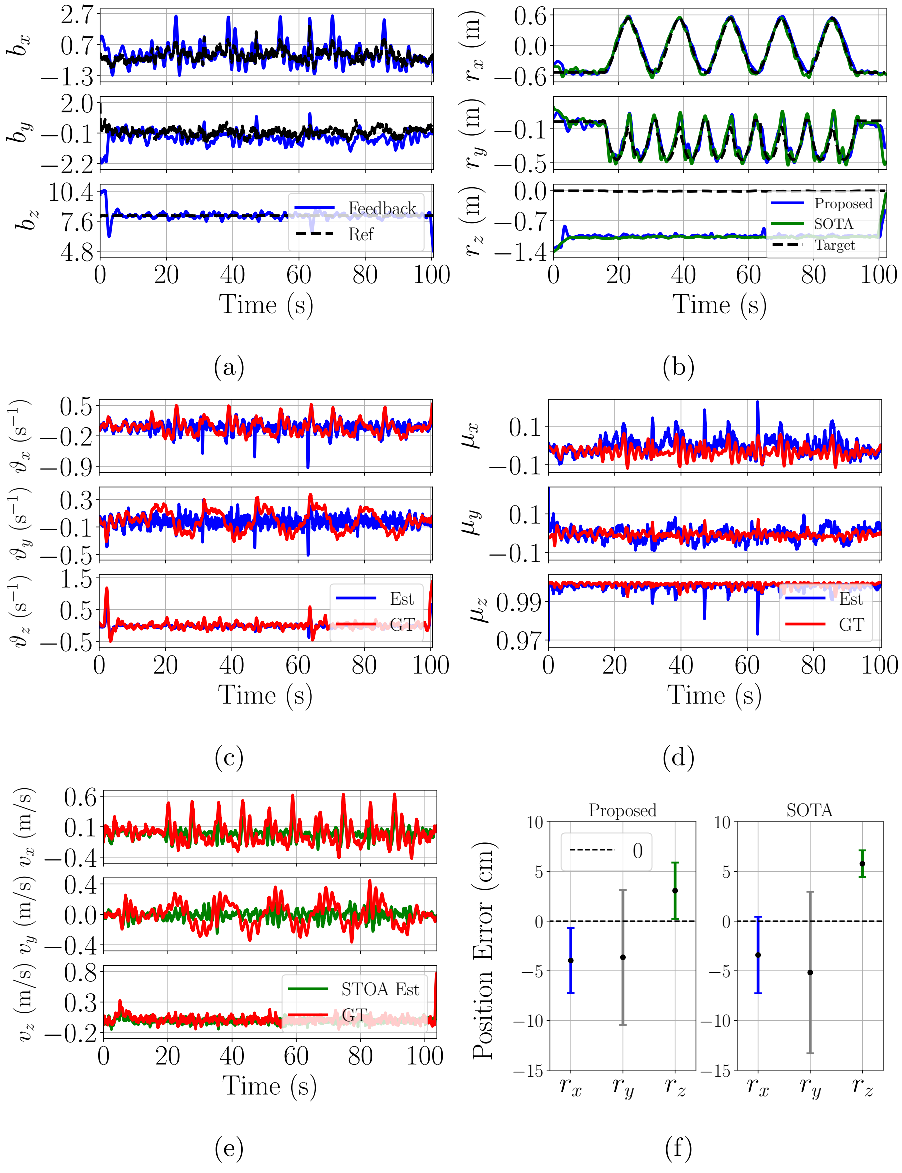
[2] https://wiki.paparazziuav.org/wiki/Main_Page

sual feedback. The camera package contains a Cortex A7 processor from ARM, AR0135 camera sensor with Field of View (FoV) of 120° from ONSemiconductor, and ICM-20948 IMU from InvenSense. The relatively large FoV decreases the chance of losing the target. The IMU outputs angular rates for the EKF estimator at 100 Hz. The four corners of ArUco fiducial marker [31] with length $l = 17.4$ cm were detected on the undistorted image at 30 Hz and tracked as visual features (four points were found to be sufficient to provide reliable tracking performance in the experiments). The corners were then re-projected onto the image plane at $z = 1$ using the intrinsic matrix of the pinhole camera model.

To speed up the detection, marker's locations from the previous image were used to narrow down the search area in the current image. Data package consisting of four detected corners, ratio velocity, plane's unit normal and desired visual feature centroid was transmitted to the motherboard of Bebop 2 through serial communication. The onboard autopilot of Bebop 2 generated the desired vehicle's attitude via the proposed IBVS controller. The onboard quaternion-based PD attitude control was implemented through Paparazzi and operated at 512 Hz for both attitude and angular rate loop to maneuver the robot to the desired attitude. For experiments, corners $\boldsymbol{p}_r$ in the desired camera frame were pre-calculated by re-projecting the points on the target to the frame for simplicity. This requires knowledge of the structure and the inclination of the target with respect to the gravity. However, the process can be replaced by taking a picture with the desired camera pose and detecting the corners in advance without knowing the orientation of the target. This is unlike the method in [4] that prior knowledge of the target's inclination was directly needed by the controller.

7.2. Flight Evaluation with a Dynamic Horizontal Target

As shown in Fig. 5 (b), a servo motor was employed to construct a moving target. The marker target was horizontally attached to an arm connected to the servo motor so that it could travel along an arc of a semicircle (radius: 50 cm) at a constant angular velocity of 0.3 rad/s (≈ 15 cm/s). To begin, the



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6: Experimental results with a horizontally moving target. (a) The visual feature centroid, (b) Position, (c) Ratio velocity, (d) Plane's unit normal, (e) Velocity from the SOTA method. (f) Mean and standard deviation of the position errors between the robot and the target for from the proposed and SOTA methods ($t = 17 - 90$ s). The ground truths (GT) are calculated from the motion capture system feedback. For (e), the position error along z-axis is obtained from the difference between the robot's altitude and desired height above the marker (1 m).

quadrotor was commanded to maneuver to an initial hovering spot about 1.4 m above the stationary marker until approximately $t = 1$ s with the motion capture feedback. Then, the IBVS controller was activated (see Fig. 5 (b) and the supplementary video). Both visual feature $\boldsymbol{b}$ and position quickly converged to the desired values and the robot remained stable above the static marker for 10 s as shown in Fig. 6 (a) and (b). The proposed line segment measurement is vital for suppressing the drift of the normal vector when the robot is motionless. After that, the target was actuated to move back and forth along the arc five times during $t = 17 - 90$ s. The robot then landed at $t = 100$ s. The tracking performance is reflected in the Fig. 6 (a)-(b) and (f). The results in Fig. 6 (a)-(b) demonstrate that the IBVS controller enables the quadrotor to converge quickly to the desired location despite the large initial deviation in position. Then, the position of the robot followed the moving object closely, the vehicle robustly retained the desired pose with respect to the target, and the error of the visual feature remains tightly bounded. This is despite the movement of the target that was not included in the model. The obtained performance manifests the robustness of the devised strategy.

Fig. 6 (c)-(d) depicts the estimation results of the normal vector and ratio velocity. It can be seen that the estimates of these states follow the ground truths closely except the ratio velocity when the target was moving. This is because of the assumption that the target remains static. Furthermore, the average computational time per frame (on Jevious camera system) was 22 ms, with the image undistortion (15.1 ms) and tag detection (6.4 ms) accounting for 98%. The numbers advocates the efficiency of the EKF framework when applied with a small number of required visual features.

To compare the proposed method against a state-of-the-art (SOTA) virtual camera-based method [4], the same experiment was repeated. To apply the SOTA method, an additional Madgwick attitude estimator [32] was implemented on Jevois camera package to deduce the robot's acceleration at 100 Hz. Fig. 6 (b) and (f) shows that the position tracking performance of the SOTA method is indistinguishable from the proposed method. The absolute velocity from the SOTA in Fig. 6 (e) remains mostly close to the ground truth but except when the target is in motion due to the static assumption (similar to our scaled velocity in Fig. 6 (c)). The outcomes confirm that the proposed method is comparable to the SOTA method but it offers benefits as an additional attitude estimator and prior knowledge of the plane's inclination are not required.

7.3. Flight Experiments with a 35°-Inclined Target

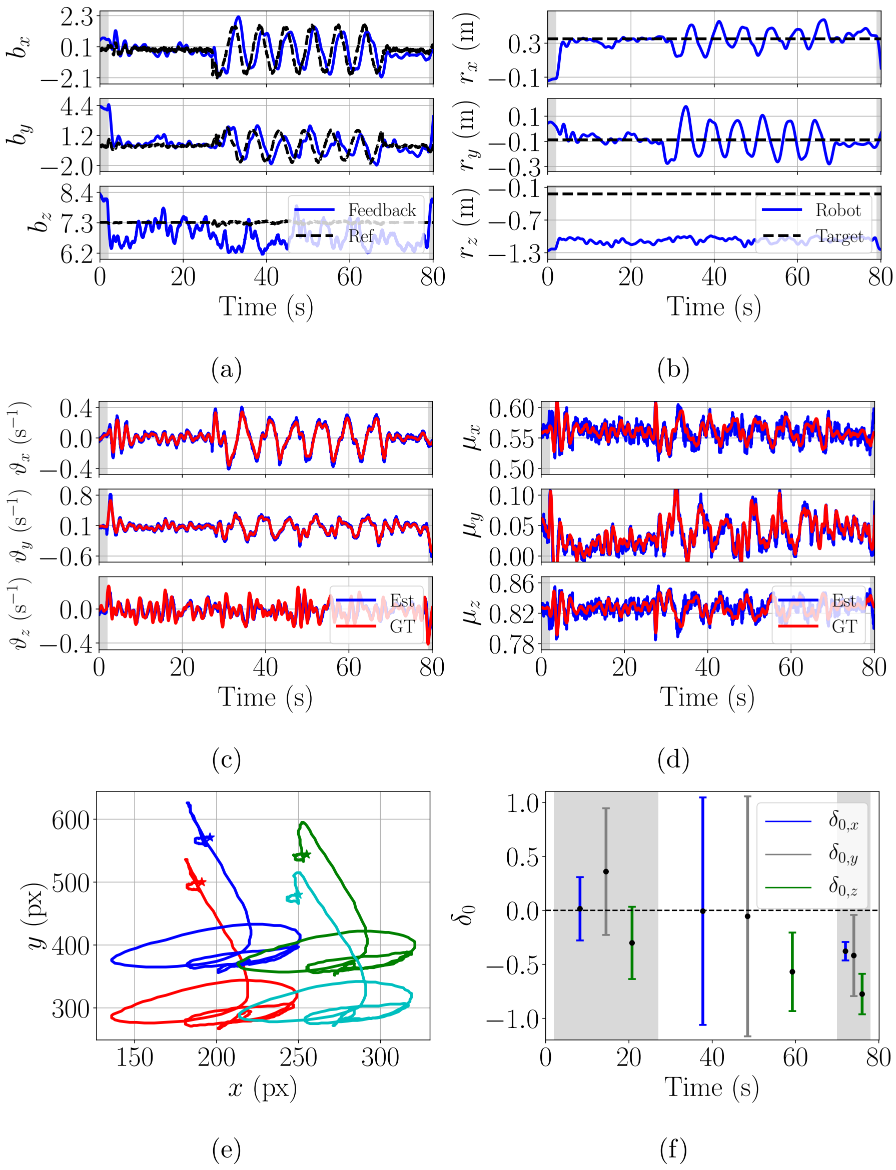
In addition to the use of ratio velocity in the formulation of IBVS controller to relax the dependence on sensors with metric scale, the regime can be employed with an inclined target plane without requiring any prior knowledge or external attitude estimator in the position control loop as needed in [4]. In this section, the proposed control and estimation framework was validated using a static target plane tilted at 35° from the horizon. Other experimental conditions remained unchanged.

As captured in Fig. 7 (a) and (b), the visual feature and position rapidly converged to the setpoint at around $t = 5$ s and remained steady above the target for 25 s. Thereafter, the control visual feature setpoint was varied according to the pre-generated trajectory to let the robot track a circular trajectory for three cycles without moving the target ($t =$ 27-70 s). The radius of the circle was 0.2 m based on the limited FoV of the camera. Fig. 7 (a) shows that the tracking error of the visual feature remained bounded. As seen in Fig. 7 (f), the error of the control feature $\boldsymbol{\delta}_0$ increased during the tracking motion since the proposed IBVS control law neglects the response time of the vehicle's low-level attitude controller.

Fig. 7 (c) and (d) plot the estimates of the ratio velocity and plane's normal against the ground-truth. The similarity between them validates the effectiveness of the proposed estimator when applied with an inclined target plane.

7.4. Outdoor Experiment

To further assess the performance of the proposed IBVS controller and estimator, the outdoor experiment, during which the quadrotor attempted to

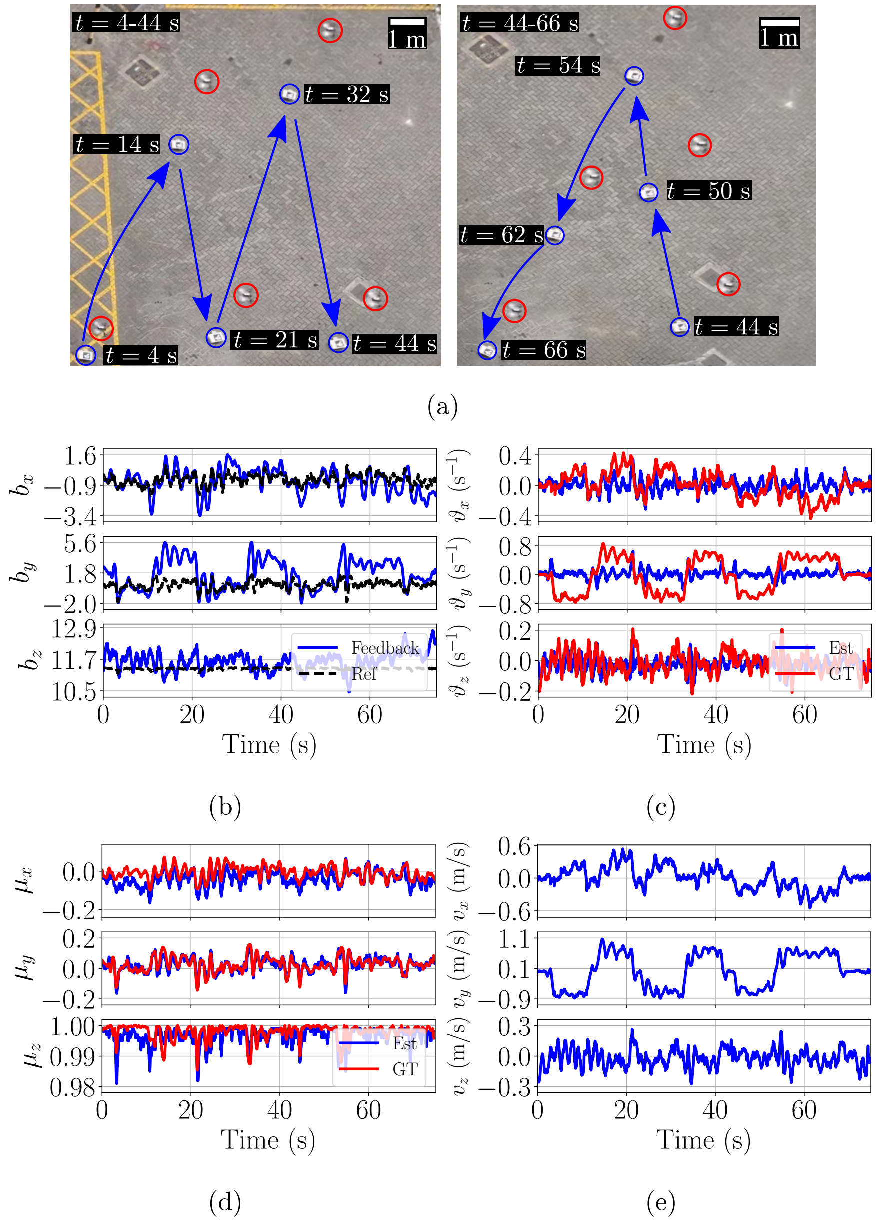


(a) (b)

(c) (d)

(e) (f)

Figure 7: Experimental results with a 35°-inclined target: time evolution of (a) the measured visual feature centroid (solid blue line) against the reference (dashed black line), (b) vehicle's position (solid blue line) against the position of the fiducial marker (dashed black line), (c) ratio velocity, and (d) plane's unit normal. (e) Trajectory of the image coordinate on the virtual image plane at $z = 1$ (for t =0-20 s. (f) Mean and standard deviation of the control feature error $\boldsymbol{\delta}_0$ during different flight stages. The tracking performance of the proposed IBVS controller is reflected in (a) and (b). The estimates of the ratio velocity and plane's unit normal from the EKF estimator are compared against the ground truths (GT) from the motion capture system. The shaded areas in Fig. (a)-(d) represent the stages during which the Bebop 2 was regulated with the feedback from the motion capture system. In (f), the shaded areas indicate the stages when the target was motionless (t =2-27 and 70-78 s).



(a)

(b)

(c)

(d)

(e)

Figure 8: Results of the outdoor experiment. (a) Composite images demonstrating the trajectory of the radio controlled car (blue circles) and Bebop 2 quadrotor (red circles) from the top. Time evolution of (b) the measured visual feature centroid (solid blue lines) against the reference (dashed black line), (c) the estimated ratio velocity against the measurement from the optical flow and range sensor (notated as GT), (d) the estimated normal vector against the measurement from the quadrotor's attitude estimator (notated as GT), and (e) the translational velocity estimated from the optic flow and range sensors.

continuously track a relatively fast moving target, was conducted. The fiducial marker was attached to a radio-controlled car (see the supplementary video). As shown in Fig. 8 (a), the car was controlled to execute an M-shaped, followed by a V-shaped trajectory. The maximum and average velocity of the car were 1.1 and 0.9 m/s, as limited by the FoV of the camera (Fig. 8 (e)). The desired relative position of the quadrotor was set to $[0, 0, -1.4]^T$ m in the maker's local frame $\mathcal{T}$. Other parameter configuration remained unchanged from the previous experiments.

Fig. 8 (b)-(d) plot the experimental results. In Fig. 8 (b), feedback of the control visual feature occasionally deviated from the reference when the car had a non-zero velocity while the robot attempted to minimize the control feature error. The reference feature varied more noticeably than in the previous experiments because of the rotation of the target around the z-axis of $\boldsymbol{R}_{rc}$ present in the definition of $\boldsymbol{\delta}_0$ from Eq. (13). The results, nevertheless, endorse the robustness of the estimation of $\boldsymbol{R}_{rc}$ despite the absence of the target's angular velocity from the measurements. Moreover, Fig. 8 (c) shows that the estimated ratio velocity remained close to the ground truth (obtained from the onboard optic flow and range sensors with certain degrees of inaccuracy, notated as GT) when the target was motionless with respect to the inertial frame. For the estimated normal vector, it could be benchmarked against the the attitude estimates from the onboard Attitude and Heading Reference System (AHRS) as the target plane is approximately horizontal. From Fig. 8 (d), one can see that the estimated normal vector follows the ground truth very closely. Overall, these results prove the robustness of the IBVS control law and estimator when the target undergoes relatively fast motion.

8. Conclusion

In this paper, we have formulated a robust and efficient virtual camera based IBVS controller and a complementary estimation scheme for small rotorcraft. The proposed IBVS control law aims at generating attitude references without the reliance on an external attitude estimator, prior knowledge of the target's inclination and metric scale sensors such as an accelerometer. This is accomplished by associating the control visual feature with the target plane and the incorporation of the ratio velocity scaled by the image moment to the IBVS formulation. To supply the required feedback, the estimator has been designed to simultaneously recover the ratio velocity, plane's normal and relative rotation between the desired and current camera frame. To verify the proposed regime, we first carried out flight experiments over a horizontal target plane. The target tracking process featured the movement of the target, proving the robustness to the designed EKF. The comparison to an established method [4] reveals a highly comparable tracking performance. The flight experiment was further conducted over an inclined target, involving the robot executing a horizontal circular trajectory over the target. The result corroborates the versatility of the framework. Lastly, the outdoor experiment was performed to continuously track the moving target involving faster speeds, zigzag path and the varying target orientation (about the vertical axis). This confirms the robustness of the proposed control law and estimator. Overall, this work offers an alternative solution for IBVS suitable for micro aerial vehicles. Future directions include the estimation of the scaled velocity of the target to enable the robot to follow a dynamic target more robustly. This could be further supplemented with a method to deal with cases where the target is out of the FoV of the camera. Similarly, motion of the target should be directly incorporated into the model to enhance the tracking performance as seen in [11, 12, 14, 13].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region of China (grant number CityU 11215220).

References

[1] T. Hamel, R. Mahony, Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach, IEEE Transactions on Robotics and Automation 18 (2) (2002) 187–198.

[2] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, L. Eck, Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle, IEEE Transactions on Robotics 25 (3) (2009) 743–749.

[3] R. Mebarki, V. Lippiello, B. Siciliano, Nonlinear visual control of unmanned aerial vehicles in gps-denied environments, IEEE Transactions on Robotics 31 (4) (2015) 1004–1017.

[4] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, X. Liang, Image-based visual servoing of a quadrotor using virtual camera approach, IEEE/ASME Transactions on Mechatronics 22 (2) (2017) 972–982.

[5] D. Lee, H. Lim, H. J. Kim, Y. Kim, K. J. Seong, Adaptive image-based visual servoing for an underactuated quadrotor system, Journal of Guidance, Control, and Dynamics 35 (4) (2012) 1335–1353.

[6] W. Zhao, H. Liu, X. Wang, Robust visual servoing control for quadrotors landing on a moving target, Journal of the Franklin Institute 358 (4) (2021) 2301–2319.

[7] J. Wu, Z. Jin, A. Liu, L. Yu, F. Yang, A survey of learning-based control of robotic visual servoing systems, Journal of the Franklin Institute.

[8] F. Chaumette, S. Hutchinson, Visual servo control. i. basic approaches, IEEE Robotics & Automation Magazine 13 (4) (2006) 82–90.

[9] M. Bloesch, M. Burri, S. Omari, M. Hutter, R. Siegwart, Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback, The International Journal of Robotics Research 36 (10) (2017) 1053–1072.

[10] T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, IEEE Transactions on Robotics 34 (4) (2018) 1004–1020.

[11] B. Hu, L. Lu, S. Mishra, Fast, safe and precise landing of a quadrotor on an oscillating platform, in: 2015 American Control Conference (ACC), IEEE, 2015, pp. 3836–3841.

[12] R. Polvara, S. Sharma, J. Wan, A. Manning, R. Sutton, Towards autonomous landing on a moving vessel through fiducial markers, in: 2017 European Conference on Mobile Robots (ECMR), IEEE, 2017, pp. 1–6.

[13] R. O. de Santana, L. A. Mozelli, A. A. Neto, Vision-based autonomous landing for micro aerial vehicles on targets moving in 3d space, in: 2019 19th International Conference on Advanced Robotics (ICAR), IEEE, 2019, pp. 541–546.

[14] N. Xuan-Mung, S. K. Hong, N. P. Nguyen, T.-L. Le, et al., Autonomous quadcopter precision landing onto a heaving platform: New method and experiment, IEEE Access 8 (2020) 167192–167202.

[15] N. Guenard, T. Hamel, R. Mahony, A practical visual servo control for an unmanned aerial vehicle, IEEE Transactions on Robotics 24 (2) (2008) 331–340.

[16] H. Jabbari, G. Oriolo, H. Bolandi, An adaptive scheme for image-based visual servoing of an underactuated uav, International Journal of Robotics and Automation 29 (1) (2014) 92–104.

[17] H. Xie, K. H. Low, Z. He, Adaptive visual servoing of unmanned aerial vehicles in gps-denied environments, IEEE/ASME Transactions on Mechatronics 22 (6) (2017) 2554–2563.

[18] D. Honegger, L. Meier, P. Tanskanen, M. Pollefeys, An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, 2013, pp. 1736–1741.

[19] S. Zhong, P. Chirarattananon, Direct visual-inertial ego-motion estimation via iterated extended kalman filter, IEEE Robotics and Automation Letters 5 (2) (2020) 1476–1483.

[20] V. Grabe, H. H. Bülthoff, D. Scaramuzza, P. R. Giordano, Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav, International Journal of Robotics Research 34 (8) (2015) 1114–1135.

[21] J. Svacha, G. Loianno, V. Kumar, Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight, IEEE Robotics and Automation Letters 4 (2) (2019) 1109–1116. doi:10.1109/LRA.2019.2894220.

[22] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, C. Silvestre, Landing of a quadrotor on a moving target using dynamic image-based visual servo control, IEEE Transactions on Robotics 32 (6) (2016) 1524–1535.

[23] H. W. Ho, G. C. H. E. de Croon, E. van Kampen, Q. P. Chu, M. Mulder, Adaptive gain control strategy for constant optical flow divergence landing, IEEE Transactions on Robotics 34 (2) (2018) 508–516. doi:10.1109/TRO.2018.2817418.

[24] H. D. Escobar-Alvarez, M. Ohradzansky, J. Keshavan, B. N. Ranganathan, J. S. Humbert, Bioinspired approaches for autonomous small-object detection and avoidance, IEEE Transactions on Robotics 35 (5) (2019) 1220–1232.

[25] P. Chirarattananon, A direct optic flow-based strategy for inverse flight altitude estimation with monocular vision and imu measurements, Bioinspiration & biomimetics 13 (3) (2018) 036004.

[26] H. Efraim, S. Arogeti, A. Shapiro, G. Weiss, Vision based output feedback control of micro aerial vehicles in indoor environments, Journal of Intelligent & Robotic Systems 87 (1) (2017) 169–186.

[27] F. Chaumette, Image moments: a general and useful set of features for visual servoing, IEEE Transactions on Robotics 20 (4) (2004) 713–723.

[28] D. Guo, K. K. Leang, Image-based estimation, planning, and control for high-speed flying through multiple openings, The International Journal of Robotics Research 39 (9) (2020) 1122–1137.

[29] O. D. Faugeras, F. Lustman, Motion and structure from motion in a piecewise planar environment, International Journal of Pattern Recognition and Artificial Intelligence 2 (03) (1988) 485–508.

[30] B. M. Bell, F. W. Cathey, The iterated kalman filter update as a gauss-newton method, IEEE Transactions on Automatic Control 38 (2) (1993) 294–297.

[31] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (6) (2014) 2280–2292.

[32] S. O. Madgwick, A. J. Harrison, R. Vaidyanathan, Estimation of imu and marg orientation using a gradient descent algorithm, in: 2011 IEEE international conference on rehabilitation robotics, IEEE, 2011, pp. 1–7.

Appendix A. Angular Velocity and Torque as Control Input

Appendix A.1. Angular Velocity as Control Input

Due to the underactuation issue of the aerial multi-rotor vehicles the desired force cannot be reached ideally. Thereby, we proceed to the third error term $\boldsymbol{\delta}_2$

to represent the difference between the desired and actual force

$$\boldsymbol{\delta}_2 = \boldsymbol{\delta}_1 - \frac{\boldsymbol{F}}{k_0 k_1}, \tag{A.1}$$

or $\boldsymbol{F} = k_0 k_1(\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2)$. Substituting Eq. (A.1) back into Eq. (19), the time derivative of $\boldsymbol{\delta}_1$ becomes

$$\dot{\boldsymbol{\delta}}_1 = - [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_1 - k_0 \boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0 \right) \boldsymbol{\delta}_1 + \frac{k_1}{mc} \boldsymbol{\delta}_2, \tag{A.2}$$

and the time derivative of $\boldsymbol{\delta}_2$ is given by

$$\begin{aligned} \dot{\boldsymbol{\delta}}_2 = & - [\boldsymbol{\omega}]_\times \boldsymbol{\delta}_2 - k_0 \boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0 \right) \boldsymbol{\delta}_1 \\ & + \frac{k_1}{mc} \boldsymbol{\delta}_2 - \frac{\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F}}{k_0 k_1}. \end{aligned} \tag{A.3}$$

Next, consider the following Lyapunov function

$$V_2 = V_1 + \frac{1}{2} \boldsymbol{\delta}_2^T \boldsymbol{\delta}_2. \tag{A.4}$$

With the Eqs. (18), (A.2) and (A.3), we have time derivative of V_2

$$\begin{aligned} \dot{V}_2 = & - k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0 \right) \boldsymbol{\delta}_1^T \boldsymbol{\delta}_1 - k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_2 + k_0 \boldsymbol{\delta}_1^T \boldsymbol{\delta}_2 \\ & + \frac{k_1}{mc} \boldsymbol{\delta}_2^T \boldsymbol{\delta}_2 - \frac{1}{k_0 k_1} \boldsymbol{\delta}_2^T \left(\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F} \right). \end{aligned} \tag{A.5}$$

Let the control input be

$$\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F} = [-\omega_y U, \omega_x U, -\dot{U}]^T, \tag{A.6}$$

according to the Eq. (3). To generate the angular velocity and thrust set points using the proposed IBVS regime, the control input is assigned to

$$\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F} := k_0 k_1 k_2 \boldsymbol{\delta}_2, \tag{A.7}$$

and Eq. (A.8) becomes

$$\begin{aligned} \dot{V}_2 = & - k_0 \boldsymbol{\delta}_0^T \boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0 \right) \boldsymbol{\delta}_1^T \boldsymbol{\delta}_1 - \left(k_2 - \frac{k_1}{mc} \right) \boldsymbol{\delta}_2^T \boldsymbol{\delta}_2 \\ & - k_0 \boldsymbol{\delta}_0 \boldsymbol{\delta}_2 + k_0 \boldsymbol{\delta}_1^T \boldsymbol{\delta}_2. \end{aligned} \tag{A.8}$$

With the fact that $\boldsymbol{a}^T\boldsymbol{b} \leq \frac{\boldsymbol{a}^T\boldsymbol{b}+\boldsymbol{b}^T\boldsymbol{b}}{2}$, we have the upper bound of $\dot{V}_2$

$$\begin{aligned}\dot{V}_2 \leq & -\frac{k_0}{2}\boldsymbol{\delta}_0\boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - 2k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_1 \\ & - \left(k_2 - \frac{k_1}{mc} - \frac{k_0}{2}\right)\boldsymbol{\delta}_2^T\boldsymbol{\delta}_2. \end{aligned} \tag{A.9}$$

To make the derivative $\dot{V}_2$ negative, the gains are properly selected as $k_0 > 0, k_1 > 2mck_0$ and $k_2 > \frac{k_1}{mc} + \frac{k_0}{2}$. Inspecting the Eqs. (A.6) and (A.7), the force error $\boldsymbol{\delta}_2$ in the lateral directions are regulated by the vehicle's roll (ω_x) and pitch (ω_y) attitude as opposed to thrust ($\dot{U}$) in the vertical direction.

Appendix A.2. Torque as Control Input

To generate the torque commands, we continue the previous procedure to obtain the new error

$$\boldsymbol{\delta}_3 = \boldsymbol{\delta}_2 - \frac{1}{k_0k_1k_2}\left(\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F}\right), \tag{A.10}$$

or $\dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times \boldsymbol{F} = k_0k_1k_2\left(\boldsymbol{\delta}_2 - \boldsymbol{\delta}_3\right)$, yielding the time derivative of $\boldsymbol{\delta}_2$

$$\begin{aligned}\dot{\boldsymbol{\delta}}_2 = & -[\boldsymbol{\omega}]_\times \boldsymbol{\delta}_2 - k_0\boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1 \\ & - \left(k_2 - \frac{k_1}{mc}\right)\boldsymbol{\delta}_2 + k_2\boldsymbol{\delta}_3, \end{aligned} \tag{A.11}$$

Time differentiating Eq. (A.10) and using Eq. (A.11), we obtain time derivative of $\boldsymbol{\delta}_3$

$$\begin{aligned}\dot{\boldsymbol{\delta}}_3 = & -[\boldsymbol{\omega}]_\times \boldsymbol{\delta}_3 - k_0\boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1 - \left(k_2 - \frac{k_1}{mc}\right)\boldsymbol{\delta}_2 \\ & + k_2\boldsymbol{\delta}_3 - \frac{1}{k_0k_1k_2}\boldsymbol{U}^v \end{aligned} \tag{A.12}$$

where the virtual control input $\boldsymbol{U}^v$ is derived using Eq. (3)

$$\begin{aligned}\boldsymbol{U}^v &= \ddot{\boldsymbol{F}} + [\dot{\boldsymbol{\omega}}]_\times \boldsymbol{F} + 2\,[\boldsymbol{\omega}]_\times \dot{\boldsymbol{F}} + [\boldsymbol{\omega}]_\times^2 \boldsymbol{F}, \\ &= [-\dot{\omega}_yU - \omega_y\dot{U}, \dot{\omega}_xU + \omega_x\dot{U}, -\ddot{U} - (\omega_x^2 + \omega_y^2)U]^T \end{aligned} \tag{A.13}$$

Now, we consider the following Lyapunov function

$$V_3 = V_2 + \frac{1}{2}\boldsymbol{\delta}_3^T\boldsymbol{\delta}_3. \tag{A.14}$$

Using Eqs. (18), (A.2), (A.11) and (A.12), the derivative of V_3 is given

$$\begin{aligned}\dot{V}_3 = &- k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_1 - k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_2 + k_0\boldsymbol{\delta}_1^T\boldsymbol{\delta}_2 \\ &- \left(k_2 - \frac{k_1}{mc}\right)\boldsymbol{\delta}_2^T\boldsymbol{\delta}_2 - k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_3 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_3 \\ &+ \frac{k_1}{mc}\boldsymbol{\delta}_2^T\boldsymbol{\delta}_3 + k_2\boldsymbol{\delta}_3^T\boldsymbol{\delta}_3 - \frac{\boldsymbol{\delta}_3^T\boldsymbol{U}^v}{k_0k_1k_2}. \end{aligned} \tag{A.15}$$

To make $\dot{V}_3$ negative, the control input is set to

$$\boldsymbol{U}^v := k_0k_1k_2k_3\boldsymbol{\delta}_3, \tag{A.16}$$

resulting in

$$\begin{aligned}\dot{V}_3 = &- k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_0 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_1 - \left(k_2 - \frac{k_1}{mc}\right)\boldsymbol{\delta}_2^T\boldsymbol{\delta}_2 \\ &- k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_2 + k_0\boldsymbol{\delta}_1^T\boldsymbol{\delta}_2 - k_0\boldsymbol{\delta}_0^T\boldsymbol{\delta}_3 - \left(\frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_3 \\ &+ \frac{k_1}{mc}\boldsymbol{\delta}_2^T\boldsymbol{\delta}_3 - (k_3 - k_2)\boldsymbol{\delta}_3^T\boldsymbol{\delta}_3, \qquad \text{(A.17)} \\ \leq &- \left(\frac{k_1}{2mc} - 2k_0\right)\boldsymbol{\delta}_1^T\boldsymbol{\delta}_1 - \left(k_2 - \frac{3k_1}{2mc} - k_0\right)\boldsymbol{\delta}_2^T\boldsymbol{\delta}_2 \\ &- \left(k_3 - k_2 - \frac{k_1}{mc} - k_0\right)\boldsymbol{\delta}_3^T\boldsymbol{\delta}_3. \end{aligned} \tag{A.18}$$

The error $\boldsymbol{\delta}_0$, $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$ and $\boldsymbol{\delta}_3$ converges to zero as long as the following condition is fulfilled

$$\begin{aligned} &k_0 > 0, \frac{k_1}{2mc} - 2k_0 > 0, \\ &k_2 - \frac{3k_1}{2mc} - k_0 > 0, k_3 - k_2 - \frac{k_1}{mc} - k_0 > 0. \end{aligned} \tag{A.19}$$

Appendix B. Derivation of the Line Segment Model

Suppose that one point $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ are visible from two camera frames: $\mathcal{C}_r$ and $\mathcal{C}_c$. The coordinates of point $\boldsymbol{P}_i$ perceived in these two frames is described by the homogeneous transformation $\boldsymbol{P}_i^r = \boldsymbol{R}_{rc}\boldsymbol{P}_i^c + \boldsymbol{t}$, where $\boldsymbol{t} \in \mathbb{R}^3$ is the camera translation between the frames. The transformation of the displacement vector $\boldsymbol{P}_i - \boldsymbol{P}_j$ between these two frames is, therefore, independent of $\boldsymbol{t}$

$$\boldsymbol{P}_i^r - \boldsymbol{P}_j^r = \boldsymbol{R}_{rc}\left(\boldsymbol{P}_i^c - \boldsymbol{P}_j^c\right). \tag{B.1}$$

Recalling the definition of the visual feature from Eq. (4), the line segments defined in Eq. (33) is rewritten as

$$\boldsymbol{l}_{ij}^{r} = \frac{\boldsymbol{P}_i^r}{d^r} - \frac{\boldsymbol{P}_j^r}{d^r} = \frac{d^c}{d^r}\boldsymbol{R}_{rc}\left(\frac{\boldsymbol{P}_i^c}{d^c} - \frac{\boldsymbol{P}_j^c}{d^c}\right) = \alpha\boldsymbol{R}_{rc}\boldsymbol{l}_{ij}^{c}. \tag{B.2}$$

with $\alpha = d^c/d^r$ or the ratio of the orthogonal distances to the two camera frames.