Unmanned Systems, Vol. 5, No. 1 (2017) 1–17

DOI: 10.1142/S2301385017500017

UNMANNED SYSTEMS
http://www.worldscientific.com/worldscinet/us

Dynamic Visual Servoing for a Quadrotor Using a Virtual Camera

Geoff Fink[*,‡], Hui Xie[*,§], Alan F. Lynch[*,¶], Martin Jagersand[†,‖]

[*]*Department of Electrical and Computer Engineering, University of Alberta*
Edmonton, Canada T6G 2R3

[†]*Department of Computing Science, University of Alberta*
Edmonton, Canada T6G 2R3

This paper presents a dynamic image-based visual servoing (IBVS) control law for a quadrotor unmanned aerial vehicle (UAV) equipped with a single fixed on-board camera. The motion control problem is to regulate the relative position and yaw of the vehicle to a moving planar target located within the camera's field of view. The control law is termed *dynamic* as it's based on the dynamics of the vehicle. To simplify the kinematics and dynamics, the control law relies on the notion of a virtual camera and image moments as visual features. The convergence of the closed-loop is proven to be globally asymptotically stable for a horizontal target. In the case of nonhorizontal targets, we modify the control using a homography decomposition. Experimental and simulation results demonstrate the control law's performance.

Keywords: Unmanned aerial vehicle; visual servoing; nonlinear control; quadrotor; experimental platform.

1. Introduction

Control of unmanned aerial vehicles (UAVs) is an active area of research [1]. These vehicles can be used in many applications such as search and rescue, infrastructure inspection, and aerial photography. Traditionally, motion control of UAVs uses vehicle position and velocity feedback from a global positioning system (GPS). However, dependence on GPS has a number of limitations. First, GPS is only available in outdoor applications with a clear view of the sky. Furthermore, GPS cannot provide the relative distance to a target of interest unless the target's GPS position is accurately known. When the target is moving, it must continuously send its GPS position to the UAV. On-board computer vision can be used to replace GPS. Vision sensors are lightweight and accurate tools which can localize the vehicle in its environment. In particular, in this paper they can provide the necessary feedback to control the vehicle's pose with respect to specific visual targets of interest.

Visual servoing methods are divided into two main approaches: position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [2]. In PBVS the pose of the vehicle is extracted from the image and used to evaluate the motion control law [3–9]. In general PBVS requires more *a priori* knowledge such as the camera calibration parameters and target geometry. On the other hand, IBVS minimizes an error function which is directly computed from features in the image plane. By avoiding the pose reconstruction in PBVS, IBVS can reduce sensitivity to camera calibration error and does not require a target model.

Traditionally, visual servoing control is divided into at least two loops [2]. The outermost loop uses vision sensor feedback to provide a velocity reference to the inner loop. It typically operates at a relatively low frequency (e.g., 30 Hz) due to the rate at which images can be acquired and processed. The inner loop regulates the velocity of the vehicle by providing force and torque commands to the vehicle at a relatively high bandwidth (e.g., 200 Hz). It is assumed that the vehicle can track the reference

Received 31 August 2015; Revised 28 September 2016; Accepted 28 September 2016; Published 8 February 2017. This paper was recommended for publication in its revised form by editorial board member, Guido de Croon.
Email Addresses: [‡]*gfink@ualberta.ca*, [§]*xie1@ualberta.ca*, [¶]*alan.lynch@ualberta.ca*, [‖]*jag@cs.ualberta.ca*

velocity perfectly and design is based on a purely kinematic vehicle model. However, the importance of including the dynamics of the vehicle is underlined in [10, 11] for both high speed tasks and underactuated systems. With respect to underactuated quadrotor UAVs, a constraint between angular displacement and linear acceleration requires us to use the vehicle's linear velocity dynamics in the control design. We refer to visual servoing that directly accounts for vehicle dynamics as *dynamic visual servoing*. Dynamic IBVS approaches for UAVs that have appeared in the literature include spherical image moment-based designs [12–16], homography-based methods [17–19], a virtual spring approach [20], virtual camera approaches [11, 21, 22], and a nonlinear generalized state transformation approach [23]. Although dynamic visual servoing has clear practical significance, few published works have thorough experimental results. Notable exceptions include [24] and [25].

This paper takes a dynamic IBVS approach to regulate the 3D position and yaw of the UAV relative to a target consisting of a point cloud on a plane. To simplify the problem we assume initially the target moves but remains horizontal. To define the desired vehicle-to-target yaw and position, an image is taken when the camera is co-planar to the target and the vehicle is at its desired configuration. This can be thought of as showing the controller its goal. The target's 3D position and yaw about its center of geometry can vary. We assume its linear velocity and yaw rate can be measured (e.g., from a motion capture system). The controller then regulates the relative position and yaw to that defined by the reference image. We provide a proof of asymptotic stability of the closed-loop for this case. To generalize this problem we also consider the problem of more general target motion in SE(3), i.e., 3D position and 3D orientation about its center. The control objective remains the same: to regulate 3D position and yaw to its desired value. In this general case, we provide simulation results but no stability proof.

This paper is organized as follows. In Sec. 2, we recall a dynamic model for the quadrotor. In Sec. 3, we present the camera model and the notion of a virtual camera. Next we provide the image feature kinematics of a moving point. We define image moments of a moving horizontal target and the corresponding kinematics. In Sec. 4, we derive the dynamic IBVS controller and prove its stability for the horizontal target case. In order to consider the nonhorizontal target case, we introduce a homography matrix in Sec. 5. In Sec. 6, we validate the control in simulation and experiment. In Sec. 7, we present our conclusions and future work. In Appendix A, we review the definition of the homography matrix and an algorithm for its decomposition.

2. Quadrotor Model

In this section, we recall an established nonlinear quadrotor model. We consider two coordinate frames which are shown in Fig. 1: the navigation frame $\mathcal{N}$ and the body frame $\mathcal{B}$. The navigation frame is assumed inertial and its origin is fixed to the surface of the earth. Its basis is the orthonormal set of vectors $\{e_1, e_2, e_3\}$ that are oriented north, east, and down, respectively. The body frame has its origin fixed to the quadrotor's center of mass and its basis $\{b_1, b_2, b_3\}$ is oriented forward, right, and down, respectively. The navigation and body frames are related by a translation $\xi^n = [\xi_1^n, \xi_2^n, \xi_3^n]^T$ and rotation $R \in \mathrm{SO}(3)$.

The rigid body dynamics expressed in $\mathcal{B}$ is

$$m\dot{v} = -m\omega \times v + F + mgR^T e_3, \tag{1a}$$

$$\dot{R} = R\,\mathrm{sk}(\omega), \tag{1b}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau, \tag{1c}$$

where $v = [v_1, v_2, v_3]^T$ is the linear velocity of the vehicle expressed in $\mathcal{B}$, $\omega = [\omega_1, \omega_2, \omega_3]^T$ is the angular velocity of the vehicle in $\mathcal{B}$, m is the mass of the vehicle, g is the acceleration due to gravity, J is the inertia matrix of the vehicle, and

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

The external force and torque acting on the vehicle are denoted by F and τ, respectively. The axis of rotation of each propeller is aligned with the b_3 axis and we have

$$F = [F_1, F_2, F_3]^T = \left[0, 0, -\sum_{i=1}^{4} f_i\right]^T, \tag{2}$$

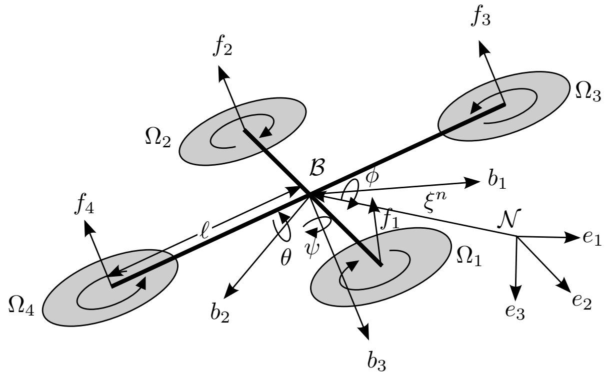


Fig. 1. Diagram of a quadrotor showing reference frames $\mathcal{N}$ and $\mathcal{B}$, Euler angles η, and direction of propeller velocities $\Omega_i > 0$.

where $f_i = k_{fi} u_i^2$ is the thrust from Propeller i, u_i is the control input normalized to $[0, 1]$ which is proportional to the speed Ω_i of Propeller i, and $k_{fi} > 0$ is a constant depending on the aerodynamics of the propeller. The position dynamics are omitted in (1) as they are replaced with the image feature kinematics that is described in Sec. 3. In this paper, we parameterize the rotation matrix by the ZYX Euler angles so that

$$R(\eta) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix},$$

where $c_\theta = \cos\theta$, $s_\theta = \sin\theta$, $\eta = [\phi, \theta, \psi]^T$, and ϕ, θ, ψ denote roll, pitch, and yaw, respectively. Then

$$\dot{\eta} = W\omega,$$

where

$$W(\eta) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \dfrac{s_\phi}{c_\theta} & \dfrac{c_\phi}{c_\theta} \end{bmatrix}$$

and $t_\theta = \tan\theta$. The quadrotor is configured as shown in Fig. 1 where the front of the vehicle is oriented between Propellers 3 and 1, and the right of the vehicle lies between Propellers 1 and 4. The torque $\tau = [\tau_1, \tau_2, \tau_3]^T$ applied to the vehicle is

$$\begin{aligned} \tau_1 &= \frac{\ell}{\sqrt{2}}(f_2 + f_3 - f_1 - f_4), \\ \tau_2 &= \frac{\ell}{\sqrt{2}}(f_1 + f_3 - f_2 - f_4), \\ \tau_3 &= -k_{\tau 1} u_1^2 - k_{\tau 2} u_2^2 + k_{\tau 3} u_3^2 + k_{\tau 4} u_4^2, \end{aligned}$$

where ℓ is the distance from the motor to the center of mass of the vehicle and $k_{\tau_i} > 0$ is an aerodynamic constant. The commonly used simple input models for F and τ are chosen to obtain a simple control law. These models could be generalized to include aerodynamic effects such as blade flapping and drag force which have been proven to be important experimentally [26, 27]. We have neglected gyroscopic torques due to variation in the angular momentum of the propellers. Further details regarding modeling can be found in [28–30].

3. Camera Models and Kinematics

3.1. *Pinhole camera*

We present a commonly used pinhole camera model found in the literature, e.g., [31, 32]. Figure 2 illustrates the model

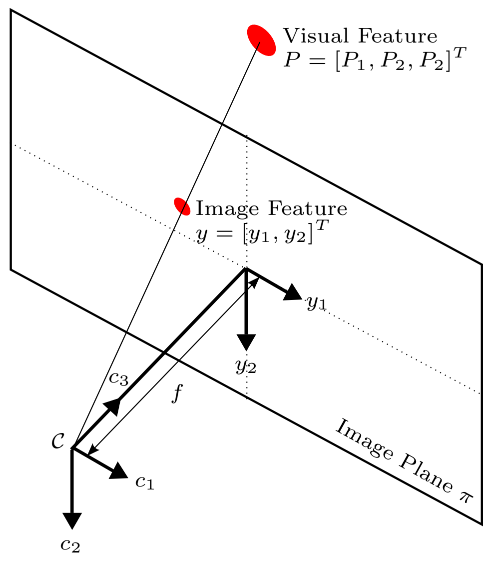


Fig. 2. Model of a pinhole camera. A visual feature point P is projected onto its image plane coordinates y in pixels.

with intrinsic parameter matrix

$$A = \begin{bmatrix} \lambda_1 & 0 & u_0 \\ 0 & \lambda_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\lambda_k = f/\rho_k, k = 1, 2$ are focal lengths in pixels, f is the focal length in meters, ρ_1, ρ_2 are the width and height of the pixel in meters, $y_0 = [u_0, v_0]^T$ is the principal point and π is the image plane.

The projection of a point $P = [P_1, P_2, P_3]^T$ in the camera frame $\mathcal{C}$ onto the image plane in homogeneous coordinates

$$Y \sim AP,$$

where $\sim$ represents equality up to a scaling of a nonzero number and $Y = [y^T, 1]^T$ is the homogeneous coordinate of y. We assume for simplicity of presentation that the frames $\mathcal{C}$ and $\mathcal{B}$ are perfectly aligned, and the camera is rigidly attached to the UAV.

To allow for moving targets we assume that point P can vary with time. We denote P^n as the position of P in $\mathcal{N}$ and assume this position is measured. Without loss of generality we assume $y_0 = 0$, and the image coordinates are

$$y = \left[\lambda_1 \frac{P_1}{P_3}, \quad \lambda_2 \frac{P_2}{P_3} \right]^T. \tag{4}$$

The single point image feature kinematics can be found by differentiating (4) with respect to time [2]. We obtain

$$\dot{y} = L_v(v - v_t) + L_\omega \omega, \tag{5}$$

where

$$L_v = \begin{bmatrix} -\dfrac{\lambda_1}{P_3} & 0 & \dfrac{y_1}{P_3} \\ 0 & -\dfrac{\lambda_2}{P_3} & \dfrac{y_2}{P_3} \end{bmatrix},$$

$$L_\omega = \begin{bmatrix} \dfrac{y_1 y_2}{\lambda_2} & -\dfrac{y_1^2+\lambda_1^2}{\lambda_1} & \dfrac{\lambda_1}{\lambda_2} y_2 \\ \dfrac{y_2^2+\lambda_2^2}{\lambda_2} & -\dfrac{y_1 y_2}{\lambda_1} & -\dfrac{\lambda_2}{\lambda_1} y_1 \end{bmatrix}$$

the linear velocity of the camera expressed in $\mathcal{C}$ is $v = R^T \dot{\xi}^n$, the angular velocity of the camera expressed in $\mathcal{C}$ is ω, and the linear velocity of the target point P expressed in $\mathcal{C}$ is $v_t = R^T \dot{P}^n$.

3.2. *Virtual camera*

A classical IBVS control for a fully actuated 6 degrees of freedom (DoF) vehicle solves for the desired six-dimensional velocity screw of the vehicle by using at least $K \geq 3$ point features. The image Jacobians in (5) for each point are stacked and the $6 \times 2K$ stacked matrix is inverted to compute the control law. Such an approach can lead to singularities and it neglects the dynamics of the vehicle. For the case of an underactuated UAV, angular and linear velocity cannot be independently specified. This prevents the classical approach from being applied. Hence, we consider a control based on a combined dynamics consisting of the image kinematics and UAV dynamics. The proposed dynamic IBVS control law uses the concept of a virtual camera to simplify the structure of the combined dynamics. The virtual camera has the same intrinsic parameters as the real camera and the same optical center, but it is oriented such that its image plane is parallel to the e_1-e_2 ground plane. That is, the origin of the virtual camera frame, denoted by $\mathcal{V}$, is fixed to the origin of $\mathcal{C}$ and moves with the vehicle. We denote the basis of $\mathcal{V}$ as $\{c_1^v, c_2^v, c_3^v\}$. The virtual and real cameras are shown in Fig. 3.

The relation between the representations of P in $\mathcal{V}$ and $\mathcal{C}$ is given by

$$P^v = [P_1^v, P_2^v, P_3^v]^T = R_{\theta\phi} P,$$

where

$$R_{\theta\phi} = \begin{bmatrix} \mathrm{c}_\theta & \mathrm{s}_\phi \mathrm{s}_\theta & \mathrm{c}_\phi \mathrm{s}_\theta \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ -\mathrm{s}_\theta & \mathrm{s}_\phi \mathrm{c}_\theta & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix}.$$

The kinematics of P^v is

$$\dot{P}^v = -\dot{\psi}\,\mathrm{sk}(e_3) P^v - v^v + v_t^v, \tag{6}$$

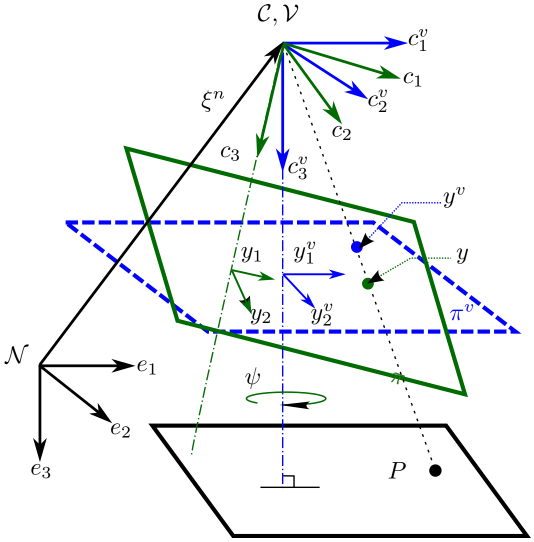


Fig. 3. The real and virtual camera frames $\mathcal{C}$ and $\mathcal{V}$, respectively. The image plane of the virtual camera π^v remains horizontal (i.e., parallel to the e_1-e_2 plane).

where $v^v = [v_1^v, v_2^v, v_3^v]^T = R_{\theta\phi} v$ is the velocity of the camera expressed in $\mathcal{V}$, $v_t^v = R_{\theta\phi} v_t$ is the target velocity expressed in $\mathcal{V}$. The corresponding projected image point y^v in the virtual camera image plane is

$$y^v = [y_1^v, y_2^v]^T = \left[\lambda_1 \frac{P_1^v}{P_3^v}, \lambda_2 \frac{P_2^v}{P_3^v} \right]^T. \tag{7}$$

The conversion from the real camera image coordinate to virtual camera image coordinate is

$$y^v = \frac{1}{\beta} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \end{bmatrix} R_{\theta\phi} \begin{bmatrix} \lambda_2 y_1 \\ \lambda_1 y_2 \\ \lambda_1 \lambda_2 \end{bmatrix}, \tag{8}$$

where

$$\beta = e_3^T R_{\theta\phi} \begin{bmatrix} \lambda_2 y_1 \\ \lambda_1 y_2 \\ \lambda_1 \lambda_2 \end{bmatrix} = \lambda_1 \lambda_2 \mathrm{c}_\phi \mathrm{c}_\theta - y_1 \lambda_2 \mathrm{s}_\theta + y_2 \lambda_1 \mathrm{s}_\phi \mathrm{c}_\theta.$$

The matrix $R_{\theta\phi}$ is assumed measured from the on-board attitude and heading reference system (AHRS). Based on (6) and (7) we can solve for the image kinematics in the virtual camera:

$$\dot{y}^v = \frac{1}{P_3^v} \begin{bmatrix} -\lambda_1 & 0 & y_1^v \\ 0 & -\lambda_2 & y_2^v \end{bmatrix} (v^v - v_t^v) + \frac{1}{\lambda_1 \lambda_2} \begin{bmatrix} \lambda_1^2 y_2^v \\ -\lambda_2^2 y_1^v \end{bmatrix} \dot{\psi}$$

An important property of these kinematics is that they are independent of roll and pitch rates. The kinematics of the real camera (5) does not have this independence. In fact, the virtual camera can be seen as defining new state coordinates in which dependence on roll and pitch rates is eliminated. The new coordinates are similar in nature to those used in our work in [23, 33] where they were also used to eliminate roll and pitch rates.

3.3. *Moment features*

To further simplify the interaction matrix we use image moments defined in [34, 35]. We assume there are $K > 1$ target point correspondences in the images denoted by $P_k, 1 \leq k \leq K$. These points make up a target which remains horizontal and can rotate about its geometric center. Figure 4 shows a target composed of four (i.e., $K = 4$) points. A frame $\mathcal{T}$ is attached to the target. The origin of $\mathcal{T}$ is at the geometric center of the target. We denote this geometric center by vector $\bar{P}^n$. The basis of $\mathcal{T}$ is $\{t_1, t_2, t_3\}$ which is defined so that the $t_1 - t_2$ plane is parallel to the $e_1 - e_2$ plane, and $t_3 = e_3$. The yaw between $\mathcal{N}$ and $\mathcal{T}$ is denoted by ψ_t and its positive direction is defined in Fig. 4. We remark that the image moments defined simplify the kinematics in the case where the target plane is parallel to the camera image plane. This is ensured by using a virtual camera frame. When the target is parallel to the camera image plane, all target points have the same depth P_3^v in $\mathcal{V}$. The image moment feature $s = [s_1, s_2, s_3, s_4]^T$ is a function of image coordinates in the virtual camera and is defined as

$$s_1 = s_3 y_{1g}^v, \tag{9a}$$

$$s_2 = s_3 y_{2g}^v, \tag{9b}$$

$$s_3 = \sqrt{\frac{\mu_{20}^* + \mu_{02}^*}{\mu_{20} + \mu_{02}}}, \tag{9c}$$

$$s_4 = \frac{1}{2} \arctan \frac{2\mu_{11}\lambda_1\lambda_2}{\lambda_2^2\mu_{20} - \lambda_1^2\mu_{02}}, \tag{9d}$$

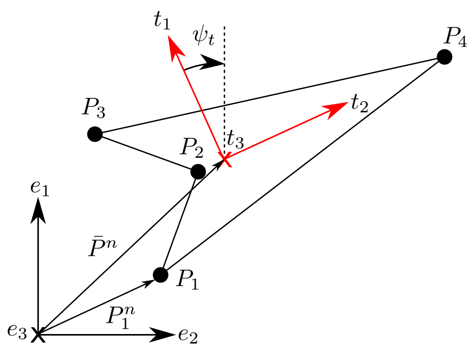


Fig. 4. Horizontal target composed of $K = 4$ points denoted $P_k, 1 \leq k \leq 4$. The position of the geometric center is denoted $\bar{P}^n = \sum_{k=1}^{K} P_k^n$, where P_k^n is P_k expressed in $\mathcal{N}$. The angle between target frame $\mathcal{T}$ and $\mathcal{N}$ is ψ_t.

where $y_{ig}^v = \frac{1}{K}\sum_{k=1}^{K} y_{ik}^v$, y_{ik}^v is the ith virtual image plane coordinate for the kth point, $\mu_{ij} = \sum_{k=1}^{K} (y_{1k}^v - y_{1g}^v)^i (y_{2k}^v - y_{2g}^v)^j$, and μ_{ij}^* is the desired value of μ_{ij} which corresponds to when the vehicle is in its desired pose. For simplicity we assume $\lambda_1 = \lambda_2$ so the image kinematics are

$$\begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{s}_3 \end{bmatrix} = \frac{-1}{P_3^{v*}} \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{v}^v + \begin{bmatrix} s_2 \\ -s_1 \\ 0 \end{bmatrix} \dot{\psi}, \tag{10a}$$

$$\dot{s}_4 = \dot{\tilde{\psi}}, \tag{10b}$$

where P_3^{v*} denotes desired depth and

$$\tilde{v}^v = R_{\tilde{\psi}}^T (v^n - \bar{v}_t^n)$$
$$\tilde{\psi} = \psi - \psi_t$$
$$R_\psi = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the velocity of the target's geometric center is

$$\bar{v}_t^v = \frac{1}{K}\sum_{k=1}^{K} v_{tk}^v$$

and v_{tk}^v is the linear velocity of the kth point in $\mathcal{V}$. The image features s_1, s_2, s_3 are used to control the translational motion of the vehicle, and s_4 controls the yaw motion. Without loss of generality, we regulate the image feature s to a desired value $s^* = [0, 0, 1, 0]^T$. This ensures that the vehicle's lateral position tracks the target's geometric center, its yaw is the same as the target, and its height above the target is P_3^{v*}. Nonzero s_1^* and s_2^* values can be chosen, but are less practically relevant since we require the target to remain in the camera's limited field of view. The values of s_3^* and s_4^* can be chosen to regulate specific desired heights and yaw. We remark in practice that the choice of $s_1^* = s_2^* = 0$ does not imply that the camera is directly above the geometric center of the target. An offset can exist due to misalignment between the body and camera frames; and the center of mass and geometric center.

4. Dynamic IBVS Control for Moving Horizontal Targets

The control of UAVs is typically divided into an inner and outer loop. We adopt this structure here as shown in Fig. 5. This inner-outer loop structure is often used independent of the sensors employed (e.g., camera or GPS). The inner loop tracks an orientation reference of the vehicle, and the outer loop provides a roll, pitch, and yaw reference to the inner

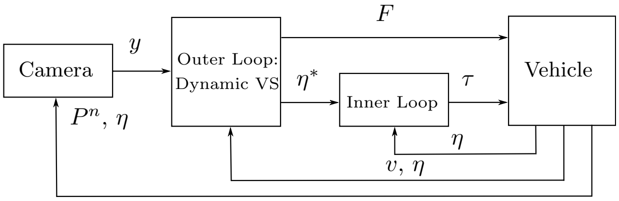


Fig. 5. Inner-outer loop Dynamic IBVS control structure.

loop in order to achieve control of translational and yaw motion. This structure arises in part from a difference in sensor frequencies and provides a practical solution to motion control found in many autopilots [36]. The multi-loop structure is useful for saturating the reference angles to avoid unsafe large roll and pitch angles. In the case of visual servoing, the decoupled control structure is natural due to the low frequency at which the camera acquires and processes images. On the other hand, the relatively fast IMU sensors driving the inner attitude control allow it to run at a higher rate. The control objective in this paper is to develop a control law that regulates the image feature s to a desired value s^*. To derive the control we make two practical assumptions. The first is a small angle approximation i.e., we assume that the roll and pitch angles are small. Hence, thrust in the virtual camera frame $\mathcal{V}$ is

$$F^v = F_3[-\mathrm{s}_\theta \mathrm{c}_\phi, \mathrm{s}_\phi, -\mathrm{c}_\theta \mathrm{c}_\phi]^T \approx F_3[-\theta, \phi, -1]^T. \tag{11}$$

This is a practical assumption as small roll and pitch is required to keep the target in the camera's field of view. The second assumption is that the inertia matrix is diagonal $J = \mathrm{diag}(J_1, J_2, J_3)$ i.e., the quadrotor's mass is symmetrically distributed.

The dynamics of the quadrotor and the target in the virtual camera frame $\mathcal{V}$ is

$$\dot{\tilde{v}}^v = -\dot{\tilde{\psi}}\mathrm{sk}(e_3)\tilde{v}^v + g e_3 + F^v/m - \bar{a}^v_t, \tag{12}$$

where $\bar{a}^v_t$ is the acceleration of the target.

Here we use the previously made assumption that the camera frame $\mathcal{C}$ and the body frame $\mathcal{B}$ are identical. The assumption is realistic in our experimental platform as we have placed the camera as close to the center of mass as possible. The frames are only offset slightly in the b_3 direction. We assume knowledge of $\dot{\psi}_t$ and $\bar{v}^v_t$ (e.g., transmitted from a friendly vehicle or measured from optical flow). To derive a control law we write the dynamics (12) with the image moment kinematics (10) as

$$\begin{aligned}
\dot{s}_1 &= -\lambda k_p \tilde{v}^v_1 + s_2\dot{\psi}, && (13a)\\
\dot{s}_2 &= -\lambda k_p \tilde{v}^v_2 - s_1\dot{\psi}, && (13b)\\
\dot{s}_3 &= -k_p \tilde{v}^v_3, && (13c)\\
\dot{\tilde{v}}^v_1 &= \tilde{v}^v_2\dot{\psi} + u_\theta, && (13d)\\
\dot{\tilde{v}}^v_2 &= -\tilde{v}^v_1\dot{\psi} + u_\phi, && (13e)\\
\dot{\tilde{v}}^v_3 &= u_F, && (13f)\\
\dot{s}_4 &= u_\psi, && (13g)
\end{aligned}$$

where $\delta_3 = s_3 - 1$, $k_p = 1/P^{v*}_3$, and the inputs are $u_\theta = -F^v_3\theta/m - \tilde{v}_2\dot{\psi}_t - \bar{a}^v_{t1}$, $u_\phi = F^v_3\phi/m + \tilde{v}_1\dot{\psi}_t - \bar{a}^v_{t2}$, $u_F = -F^v_3/m + g - \bar{a}^v_{t3}$, and $u_\psi = -\dot{\tilde{\psi}}$.

Theorem 4.1. *Subsystem (13a)–(13f) is globally asymptotically stable and subsystem (13g) is globally exponentially stable with input*

$$\begin{aligned}
u_F &= k_{hd} v^v_3 - k_{hp}\delta_3 - k_{hi}\int_0^t \delta_3(\epsilon)\mathrm{d}\epsilon, && (14a)\\
u_\theta &= -k_{\ell 2}(\tilde{v}^v_1/k_{\ell 1} - s_1), && (14b)\\
u_\phi &= -k_{\ell 2}(\tilde{v}^v_2/k_{\ell 1} - s_2), && (14c)\\
u_\psi &= k_\psi s_4, && (14d)
\end{aligned}$$

where k_{hp}, k_{hd}, k_{hi}, $k_{\ell 1}$, $k_{\ell 2}$, and k_ψ are positive constant gains such that $k_{hi} < k_{hp}k_{hd}$ and $P^{v*}_3 k_{\ell 2} > k^2_{\ell 1}$.

Proof. First we consider the height motion subsystem (13c) and (13f) and we define the error signal

$$\delta_3 = s_3 - 1.$$

In addition, we augment the state by with an integral

$$\zeta = \int_0^t \delta_3(\epsilon)\mathrm{d}\epsilon.$$

Let $u_F = k_{hp}\delta_3 - k_{hd}\dot{\tilde{v}}^v_3 + k_{hi}\zeta$ then the closed loop dynamics are

$$\begin{bmatrix} \dot{\delta}_3 \\ \dot{\tilde{v}}^v_3 \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & -k_p & 0 \\ k_{hp} & -k_{hd} & k_{hi} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_3 \\ \tilde{v}^v_3 \\ \zeta \end{bmatrix}.$$

Then if $k_{hp}, k_{hd}, k_{hi} > 0$ and $k_{hi} < k_p k_{hp} k_{hd}$ the closed loop origin of the height motion subsystem is globally exponentially stable. Next, we consider the translational motion subsystem (13a), (13b), (13c), and (13d). Define error variables

$$\begin{aligned}
\delta_1 &= \frac{1}{\lambda}s_1\\
\delta_2 &= \frac{1}{\lambda}s_2\\
\delta_4 &= \tilde{v}^v_1/k_{\ell 1} - \delta_1\\
\delta_5 &= \tilde{v}^v_2/k_{\ell 1} - \delta_2
\end{aligned}$$

and consider the Lyapunov function candidate

$$V = \frac{1}{2}\delta_1^2 + \frac{1}{2}\delta_2^2 + \frac{1}{2}\delta_4^2 + \frac{1}{2}\delta_5^2$$

then

$$\dot{V} = (u_\theta \delta_4 + u_\phi \delta_5)/k_{\ell 1} + k_{\ell 1} k_p (-\delta_1^2 - \delta_2^2 + \delta_4^2 + \delta_5^2).$$

Letting $u_\theta = -k_{\ell 2}\delta_4$ and $u_\phi = -k_{\ell 2}\delta_5$ then

$$\dot{V} = -k_{\ell 1} k_p (\delta_1^2 + \delta_2^2) - (k_{\ell 2} - k_p k_{\ell 1}^2)(\delta_4^2 + \delta_5^2)/k_{\ell 1}.$$

If $k_{\ell 2} > k_{\ell 1}^2 k_p$ then $\dot{V}_2$ is negative definite. Therefore, the translational subsystem (13a)–(13f) is globally asymptotically stable. For the yaw subsystem (13g) we substitute the control law (14d) into the dynamics (10b) and obtain

$$\dot{s}_4 = -k_\psi s_4$$

which is globally exponentially stable. □

The control law requires an initialization and main phase. In the initialization phase the quadrotor is manually hovered so that the target is in the camera's field of view. The user then selects the target, indicates its orientation in the camera frame and the desired distance to the target. Note that the desired distance to the target is not specified in meters, but is an image feature. That is, the user selects the desired scale in the image: larger (closer), the same (maintain distance), or smaller (further away). This feature can be changed at anytime during flight.

5. Nonhorizontal Targets

In previous work [11, 25] it was assumed that the planar target was horizontal. This assumption was made so that the concept of a virtual camera as used in [25] could be applied. If the virtual camera is applied to nonhorizontal targets it has been shown that steady state error results in the image features, especially in the third component s_3 which is related to relative height. In this paper, we compensate for nonhorizontal target planes using a homography decomposition which can be used to estimate the orientation between the camera image plane and target plane.

5.1. *Applying the homography*

After computing the homography decomposition as described in Appendix A we use it to define a new virtual camera frame $\mathcal{V}'$. The following discussion explains how this new frame compensates for nonhorizontal targets. The following assumptions are made to provide a precise description of the pose of $\mathcal{V}'$ without having to know the geometry of the target and the relative pose from the camera to the target.

Assumptions:

(1) The target rotates about its geometric center.
(2) The target plane in the desired image is parallel to the camera plane, i.e., $n^1 = [0, 0, 1]^T$.
(3) The camera is centered laterally above the target, i.e., the geometric center of the target lies on the c_3 axis.

The frame $\mathcal{V}'$ has the same lateral position and orientation relative to the target as the camera to the desired scene. That is, only the height of the camera can differ relative to the desired case. Using $\mathcal{V}'$ we can estimate the area of the target independent of the target and camera's orientation. We remark that virtual camera $\mathcal{V}$ introduced in Sec. 3.2 remains parallel to the ground. Using a combination of $\mathcal{V}$ and $\mathcal{V}'$ we can define image features that behave as if the target is always parallel to the ground.

Figure 6 illustrates the idea when relative motion is restricted to 1 DoF in the c_3 direction, and the target plane can rotate by an angle θ. The 3D case is restricted to relative camera motion in the c_3 direction, and the target can rotate in 2 rotational DoF. The extension of the 2D case to the 3D case is straightforward. In the figure an initial desired image Y^1 is taken from the camera $\mathcal{C}^1$ of the target in its desired position (green). Next, suppose the target rotates about its geometric center (denoted by a red "x") by an angle θ. At this point if another image were taken with $\mathcal{C}^1$ then the length (area in 3D) of the target in the image will have decreased. Given this condition, the control law would attempt to lower the camera to increase the length of the target. However, the control objective is to regulate the height above the target independent of the target's orientation. To solve this problem a homography matrix H is estimated between the desired image (camera $\mathcal{C}^1$ with a horizontal target) and the current image (camera $\mathcal{C}^1$ with a nonhorizontal target). Next, the homography is decomposed into the rotation and scaled translation of the camera. Since the decomposition and estimation of the homography assumes the target is static and only the camera moves, we can effectively rotate the camera about the target's

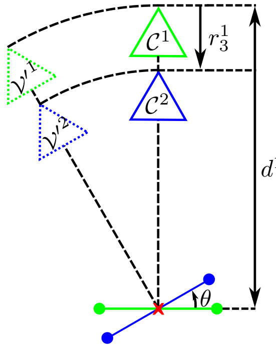


Fig. 6. An initial desired image is taken from $\mathcal{C}^1$ of the horizontal target. The virtual camera associated with $\mathcal{C}^1$ and the rotated target is $\mathcal{V}'^1$. The virtual camera associated with $\mathcal{C}^2$ and the rotated target is $\mathcal{V}'^2$.

geometric center by an angle θ to $\mathcal{V}'^1$. An image taken from virtual camera $\mathcal{V}'^1$ will show the same target length as the desired image and thus the control law will not command the camera to move. If at some point in time the camera were to move to $\mathcal{C}^2$ and take an image of the nonhorizontal target, the length of the target in the vehicle could be exactly the same as its length in the desired image. A control law based on the original virtual camera $\mathcal{V}$ would incorrectly not command the camera to move upward. A homography is calculated between the image taken in $\mathcal{C}^1$ of the horizontal target and the image in $\mathcal{C}^2$ of the nonhorizontal target. This homography is decomposed to define the virtual camera $\mathcal{V}'^2$. The length of the target in $\mathcal{V}'^2$ will be less than in the desired image and the camera will be commanded to rise.

The image $Y' \in \mathbb{R}^3$ in the virtual camera frame $\mathcal{V}'$ is approximately given by

$$Y' \sim A \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 - \left(\frac{1}{d^1} r_3^1\right)^2 \end{bmatrix} A^{-1} Y^*, \tag{15}$$

where $Y^* \in \mathbb{R}^3$ denotes the image in the desired camera frame. The scaled displacement $\frac{1}{d^1} r_3^1$ is obtained from the homography decomposition as explained in Appendix A. Relation (15) is equivalent to

$$y^{v'} = \frac{1}{1 - \left(\frac{1}{d^1} r_3^1\right)^2} y^*, \tag{16}$$

where $y^{v'} \in \mathbb{R}^2$ denotes the image in $\mathcal{V}'$, and $y^* \in \mathbb{R}^2$ is the desired image.

Motivated by the above discussion, we introduce a modified feature

$$s_3' = \sqrt{\frac{\mu_{20}'^* + \mu_{02}'^*}{\mu_{20}' + \mu_{02}'}} \tag{17}$$

which simplifies to

$$s_3' = \sqrt{\frac{\sum_{k=1}^{K} (y_{1k}^{v'*})^2 + (y_{2k}^{v'*})^2}{\sum_{k=1}^{K} (y_{1k}^{v'})^2 + (y_{2k}^{v'})^2}}$$

since $y_{1g}^{v'} = y_{2g}^{v'} = 0$ by assumption. The kinematics of the feature is

$$\dot{s}_3' = \frac{-v_3^{v'}}{P_3^{v'*}}. \tag{18}$$

Both virtual camera frames $\mathcal{V}$ and $\mathcal{V}'$ rely on assumptions. That is, $\mathcal{V}$ assumes that the target is horizontal, but makes no assumptions about the pose of the camera. On the other hand $\mathcal{V}'$ allows for nonhorizontal targets, but assumes the camera is constrained to move vertically above the target. In order to exploit the benefits of both virtual frames, we combine the area moment features into one feature $\tilde{s}_3$. The resulting image feature vector $\tilde{s}$ is defined as

$$\begin{aligned} \tilde{s}_1 &= s_3' s_1 / s_3, \\ \tilde{s}_2 &= s_3' s_2 / s_3, \\ \tilde{s}_3 &= \alpha s_3 + (1-\alpha) s_3', \\ \tilde{s}_4 &= s_4, \end{aligned} \tag{19}$$

where $\alpha \in [0, 1]$ is a tuning parameter. In practice we have found that an equal weighting (i.e., $\alpha = 1/2$) of the two features to yield good results. The algorithm is summarized in Algorithm 1.

Algorithm 1 Dynamic Visual Servoing

function MAIN LOOP
  Takeoff
  Hover
  while User wishes to track object **do**
    $\{y^1, n^1, a^*\} \leftarrow$ Ini()
    Track($\{y^1, n^1, a^*\}$)
  end while
  Land
end function

Initialization Phase:

function INI
  $y^1 \leftarrow$ User select target in image
  $n^1 \leftarrow$ User select orientation of target in image
  $a^* \leftarrow$ User selects desired distance of target
end function

Main Phase:

function TRACK($\{y^1, n^1, a^*\}$)
  $y \leftarrow$ Take Image ▷ Eq. (4)
  $y^2 \leftarrow$ Track points in image(y)
  $H \leftarrow$ Estimate Homography(y^1,y^2)
  $R, \frac{1}{d^1} r^1 \leftarrow$ Decompose Homography (H,n^1)
  $R_{\theta\phi} \leftarrow$ Get attitude from AHRS
  $s \leftarrow$ Calc Image Features(y^2,$R_{\theta\phi}$) ▷ Eq. (9)
  $s' \leftarrow$ Estimate target area(y^1,$\frac{1}{d} r^1$) ▷ Eq. (15)
  $\tilde{s} \leftarrow$ Update Image Features(s,s') ▷ Eq. (19)
  F^b,$\eta^* \leftarrow$ Outer Loop($\tilde{s}$,v^v,v^{v*},$\dot{\psi}^*$) ▷ Eq. (14)
  $\tau \leftarrow$ Inner Loop(η,η^*,ω) ▷ Eq. (20)
  Repeat
end function

6. Validation of Dynamic IBVS Control

Physical experiments and simulations were performed to validate the proposed controllers. Due to limitations with the experimental test stand we restrict validation to the horizontal target case in experiment. In simulation we validate the nonhorizontal target case. We use the quadrotor and camera parameters in Table 1. In both experiment and simulation we used a proportional-integral-derivative (PID) inner loop control for attitude:

$$\begin{aligned} \tau_1 &= k_{p\phi} e_\phi + k_{i\phi} \int_0^t e_\phi(\tau) d\tau + k_{d\phi} \dot{e}_\phi, \\ \tau_2 &= k_{p\theta} e_\theta + k_{i\theta} \int_0^t e_\theta(\tau) d\tau + k_{d\theta} \dot{e}_\theta, \\ \tau_3 &= k_{p\psi} e_\psi + k_{i\psi} \int_0^t e_\psi(\tau) d\tau + k_{d\psi} \dot{e}_\psi, \end{aligned} \quad (20)$$

where $e_\phi = \phi^* - \phi, e_\theta = \theta^* - \theta, e_\psi = \int_0^t \dot{\tilde{\psi}}(\epsilon) d\epsilon$, and ϕ^* and θ^* denote reference angles provided by the outer loop control. The control law (20) combined with u_F determine the propeller velocities based on the model in Sec. 2.

6.1. *Experimental validation*

6.1.1. *Experimental platform*

In order to efficiently perform experimental research on nonlinear control and visual servoing of UAVs, we developed the indoor Applied Nonlinear Control Lab (ANCL) quadrotor platform. The platform includes a Vicon motion capture system that can calculate the pose of the vehicle at up to 200 Hz with millimeter accuracy. In this paper, we use the Vicon system to provide the vehicle an estimate of its velocity. The quadrotor weighs about 1.6 kg including the battery and measures about 25 cm from the center of the vehicle to the motor shafts. The custom developed quadrotor is shown in Fig. 7. We use a PX4 autopilot which is an open-hardware and open-source project [37]. The computer vision system is a CMUcam5 Pixy which is a low cost vision sensor [38]. It has a focal length of $f = 2.8$ mm, an image sensor size of 1/4" and a maximum resolution of 1280 $\times$ 800 pixels. It can find hundreds of objects in each frame at up to 50 frames per second. The image sensor is mounted on the bottom of the quadrotor and connected to the PX4 via a serial connection. Currently, it processes a 640 $\times$ 400 pixel image to find image blobs at a rate of about 50 Hz. It then transmits an identification number of the blob based on its color and the center, width, and height of the blob. More information about the platform can be found in [23].

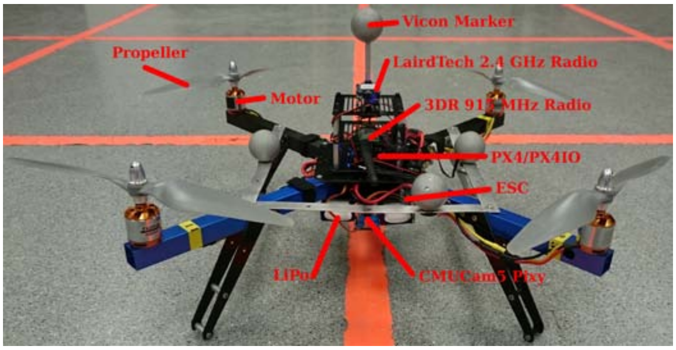


Fig. 7. The ANCL quadrotor.

Table 1. Quadrotor and camera model parameters.

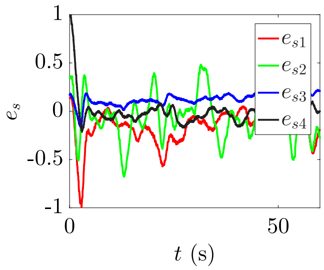
m	1.6 kg
J	diag(0.03, 0.03, 0.05) kg $\cdot$ m^2
f	2.8 mm
(ρ_1, ρ_2)	(2.5, 3.0) μm
Resolution	640 $\times$ 400 pixels
ℓ	0.25 m

6.1.2. *Experimental results for a horizontal target*

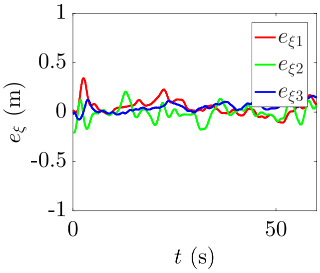
Due to limitations with the ANCL platform, experimental testing was restricted to the horizontal target case. That is, we use the proposed control law (14) which is based on a horizontal target assumption. We used the same controller gains as in Table 2. In the experiment two visual markers were placed on the ground 36 cm apart. The quadrotor was placed between the two markers and manually flown to about $[0.1, 0.1, -1.0]$ m to ensure that the markers were in the camera's field of view and to be above any ground effects. The initial yaw angle was about 50°. At this time the dynamic IBVS controller was turned on and run for about 60 s. The experimental results are shown in Fig. 8. Figure 8(a) shows the components of error in the image features

Table 2. Parameters used in simulation and experiment.

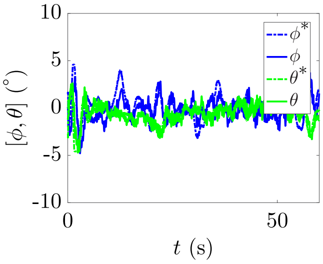
$[k_{hp}, k_{hd}, k_{hi}]$	$[4.4, 1.0, 1.0]$
$[k_{\ell 1}, k_{\ell 2}]$	$[1.2, 0.4]$
k_ψ	0.60
α	0.5
$[k_{p\phi}, k_{i\phi}, k_{d\phi}]$	$[10, 1, 3]$
$[k_{p\theta}, k_{i\theta}, k_{d\theta}]$	$[10, 1, 3]$
$[k_{p\psi}, k_{i\psi}, k_{d\psi}]$	$[6, 1, 3]$
k_f	$18 - 0.02t$ N
$\xi^n(0)$	$[0.25, 0.25, -1.0]$ m
ξ^{n*}	$[0, 0, -1.5]$ m
$\psi(0)$	45°
$\psi_t(t)$	0°



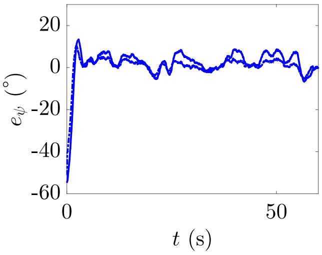
(a) Image feature error e_s



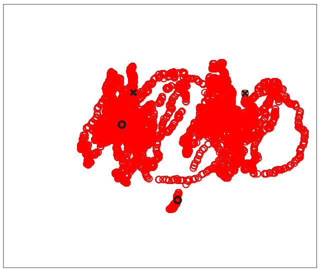
(b) Position error e_ξ



(c) Roll ϕ and pitch θ



(d) Yaw error $\tilde{\psi}$



(e) Image feature trajectory. The initial and final positions of the feature are denoted by a circle and a cross, respectively.

Fig. 8. Experimental results of the proposed dynamic IBVS control law for a horizontal target.

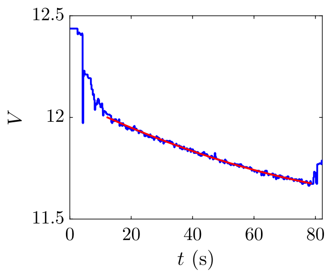
$e_s = s^* - s$ rapidly converging to a bounded trajectory near 0. We remark that the oscillations in s_1 and s_2 are likely caused by tracking error in the inner loop control, and the performance is comparable with our visual servoing experiments on the same platform in [23]. The error s_4 slowly converges to zero. Figure 8(b) shows convergence of error in vehicle position $e_\xi = [e_{\xi 1}, e_{\xi 2}, e_{\xi 3}] = \xi^{n*} - \xi^n$ to a bounded trajectory near zero.

Table 3 shows the mean and standard deviation of the error signals. The magnitude of error is similar to our past work [23] where only the two lateral DoF were controlled. Here, we control all four DoF (i.e., 3D position and yaw). Figure 8(c) shows trajectories for roll and pitch. In Fig. 8(d) the error in yaw ψ converges to zero. Finally, in Fig. 8(e) the trajectories of the image point features are shown in the image plane in pixels. The image features at time $t = 0$ s are denoted by a circle and at time $t = 60$ s by a cross.

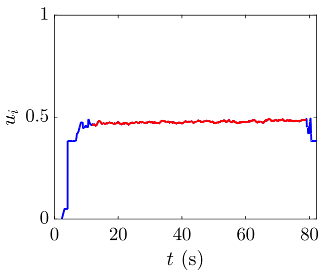
Table 3. Mean and standard deviation of error for the experimental data.

Variable	Mean	Standard deviation
e_{s_1}	0.12	0.15
e_{s_2}	0.06	0.22
e_{s_3}	0.12	0.04
e_{s_4}	0.05	0.06
$e_{\xi 1}$ (cm)	3.8	6.8
$e_{\xi 2}$ (cm)	-0.2	7.1
$e_{\xi 3}$ (cm)	-5.4	4.2
e_ψ (°)	2.9	3.7

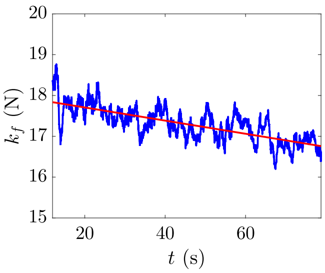
As seen in Figs. 8(a) and 8(b), the height of the vehicle drops about 10 cm over a period of about one minute. This is expected as the integral term in the control law (14a) was disabled (i.e., $k_{hi} = 0$). As the battery voltage drops, less thrust is generated for the same control signal. Although the control signal increases slightly due to the increased error, it is insufficient to compensate for the decrease in thrust coefficient. Figures 9(a) and 9(b) show battery voltage and thrust input u_i, respectively. To estimate the thrust constant in force model (2) we assume that each propeller generates the same thrust and denote thrust coefficient $k_f = k_{fi}, i = 1, 2, 3, 4$. We also assume the quadrotor is hovering. In this case the thrust coefficient is given by



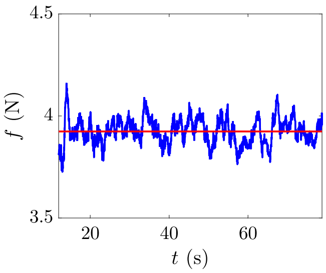
(a) Drop in battery voltage



(b) Control signal u_i



(c) Estimated thrust coefficient (red) and data (blue)



(d) Estimated force f_i (blue) and hover force $mg/4$ (red)

Fig. 9. Time varying thrust constant.

$k_f = 0.25mg/u_i^2$. We model the dependence of k_f on time as linear:

$$k_f(t) = k_{f1}t + k_{f0}. \tag{21}$$

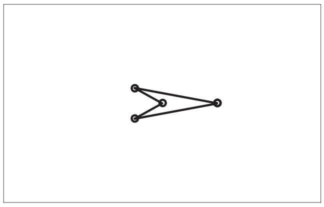
A least squares fit for the thrust coefficients in model (21) is performed. This model is shown in Fig. 9(c). Figure 9(d) shows estimated force f_i compared to the assumed hover force $mg/4$. In simulation we used (21) to show the integral term in (14a) can compensate for a decrease in thrust coefficient.

6.2. *Simulation results*

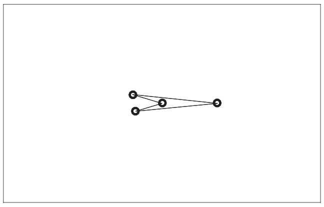
In the nonhorizontal target case, complexity of the control law increases given the homography decomposition is computed. At its current stage of development, these computations could not be implemented on the ANCL quadrotor platform. Hence, a Simulink simulation was used for validation. The simulations include the rigid body dynamics and kinematics (1), the pinhole camera (4), the virtual cameras (8) and (16), the image features (9) and their modifications (19). The algorithm used is described in Algorithm 1. We chose the visual target to be a "car" which is represented by four points in the shape of an arrow that points in the forward direction of the car. There is no requirement to know additional information about the target's geometry such as its dimensions. We simulate a control assuming the target acceleration is zero. This avoids the requirement of measuring this quantity and makes the control more practical.

In the first simulation the quadrotor was initialized directly above the stationary car that was rotated by one radian (57°) about its roll axis. The control objective was to regulate height above the car and the proposed method is compared with that previously developed in [25]. The camera images for the simulation are given in Fig. 10. The motion is such that initial (shown as circle) and final images (shown as cross) are almost identical. We observe that in both the initial real camera image [Fig. 10(b)] and the trajectory of the virtual camera image for $\mathcal{V}$ [Fig. 10(c)], the car appears compressed in the vertical axis of the image due to the inclined target. This leads to a reduced area which is measured by moment feature s_3 and thus the quadrotor flies closer to the car to increase the target's area in the image. Both the desired image [Fig. 10(a)] and the image in the new virtual camera $\mathcal{V}'$ are identical in this case.

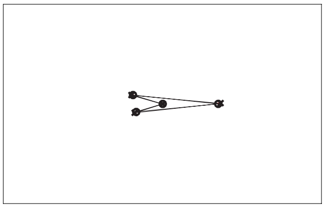
Figure 11 shows height trajectories for a number of cases. Figure 11(a) compares height using the proposed method (green) and the previous approach (blue). The desired height is 1.5 m, and the proposed algorithm provides a smaller steady state error relative to the previous work. Figure 11(b) shows the effect of time varying thrust coefficient when an integrator is not used in the control ($k_{hi} = 0$). In this case height for both of the algorithms drops significantly. Figure 11(c) shows the result with the integrator enabled ($k_{hi} = 1$). Figure 12 shows the result of the homography decomposition with roll and pitch estimated to less than one degree of error.



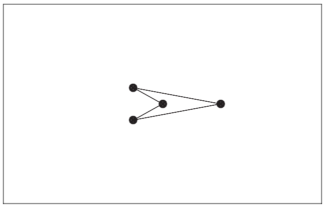
(a) Desired camera image



(b) Initial real camera image

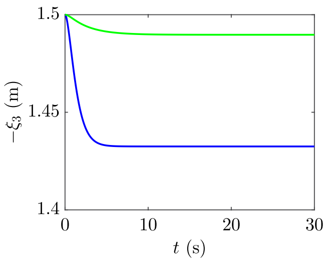


(c) Virtual camera $\mathcal{V}$ image

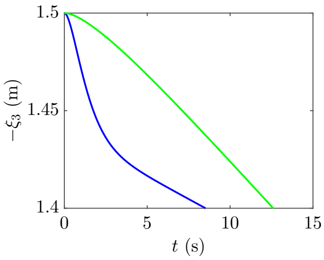


(d) New virtual camera $\mathcal{V}'$ image

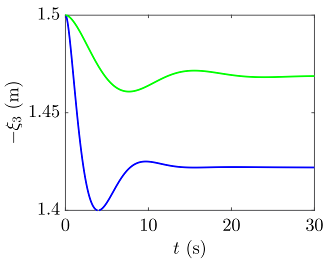
Fig. 10. Simulated camera images using the proposed method and the approach in [25]. The control objective is to regulate height above the target.



(a) Ideal battery

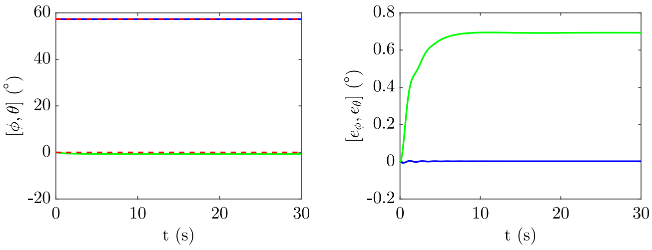


(b) Real battery without integrator compensation



(c) Real battery with integrator compensation

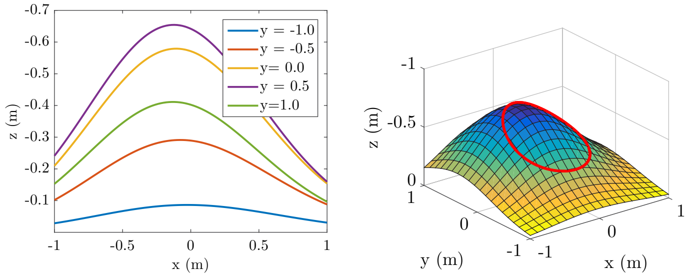
Fig. 11. Simulated height of the camera using the proposed method (green) and the previous approach in [25] (blue).



(a) Roll and pitch estimates (solid red) and actual values (dashed red)

(b) Error in estimation

Fig. 12. Roll (blue) and pitch (green) estimation using a homography decomposition.



(a)

(b)

Fig. 13. The ground plane and the trajectory of the car (red).

In the second simulation we define a surface on which the car travels:

$$z(x,y) = c_1 e^{\frac{-(x-x_{01})^2}{c_{x1}} + \frac{-(y-y_{01})^2}{c_{y1}}} + c_2 e^{\frac{-(x-x_{02})^2}{c_{x2}} + \frac{-(y-y_{02})^2}{c_{y2}}}$$

The car's path $\bar{P}^n(t)$ is defined by

$$\bar{P}^n_1(t) = \begin{cases} \dfrac{rt}{t_0} & 0 \le t < t_0 \\ r\cos(t/c_t - t_0) & t \ge t_0 \end{cases}$$

$$\bar{P}^n_2(t) = \begin{cases} 0 & 0 \le t < t_0 \\ r\sin(t/c_t - t_0) & t \ge t_0 \end{cases}$$

$$\bar{P}^n_3(t) = z(\bar{P}^n_1(t), \bar{P}^n_2(t))$$

The car's attitude is such that it is level to the tangent of the surface z. Its yaw angle is zero until after the car completes one full circle then it follows a sine wave. The attitude of the car is given by

$$\phi(x,y,t) = 2\left(\frac{x(t)-x_{01}}{c_{x1}} + \frac{x(t)-x_{02}}{c_{x2}}\right) z(x(t),y(t))$$

$$\theta(x,y,t) = -2\left(\frac{y(t)-y_{01}}{c_{y1}} + \frac{y(t)-y_{02}}{c_{y2}}\right) z(x(t),y(t))$$

$$\psi(t) = \begin{cases} 0 & 0 \le t < t_1 \\ \sin(t/c_{t\psi} - t_1) & t \ge t_1 \end{cases}$$

The surface z and the car's trajectory $[c_x, c_y, c_z]^T$ (red) is shown in Fig. 13. The constants used are in Table 4.

We ran the car following simulation using the previous algorithm where the velocity of the target is assumed to be zero and the target is assumed horizontal [25] and the proposed algorithm where the velocity of the target is no longer assumed to be zero and the target can be non-horiztontal. The result of the simulations are shown in Figs. 14 and 16.

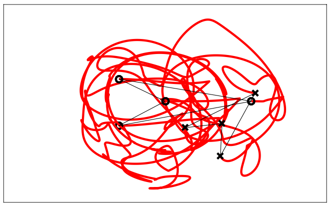
Figure 14 shows relatively poor performance of the previous method. From Fig. 14(a) the point features of the car in the image exhibit large variation, and this causes large error in the moment image features as shown in Fig. 14(c). Figure 14(f) shows loss of yaw tracking for $t \ge t_1$. Although in Fig. 14(d) the vehicle shows bounded lateral position tracking error, the magnitudes of error are large. The nonzero tracking error is from assuming the target velocity is zero.

Table 4. Simulation parameters used to define ground plane and car trajectory.

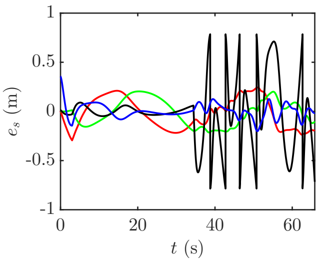
Parameter	Value
c_1	-0.5
c_2	-0.2
(x_{01}, y_{01})	$(-0.2, 0.4)$
(y_{02}, y_{02})	$(0.2, 0.2)$
c_{x1}	0.7
c_{y1}	0.8
c_{x2}	1
c_{y3}	1
c_t	5
$c_{t\psi}$	$\frac{2\pi}{5}$
t_0	3
t_1	$2\pi c_t + t0$
r	0.5

Figure 16 shows increased tracking performance using the proposed method. We observe from Fig. 16(a) that the point features of the car exhibit small variation, and image feature error is reduced as shown in Fig. 16(c). Figures 16(d) and 16(f) show good tracking performance for lateral position and yaw. Figure 15(a) shows the roll and pitch computed from the homography decomposition versus actual roll and pitch. Figure 15(b) gives the error trajectories for these angles, and we observe magnitudes less than about 5°. Another measure of error for a homography matrix is the

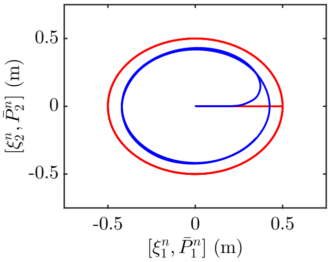
(a) Camera image



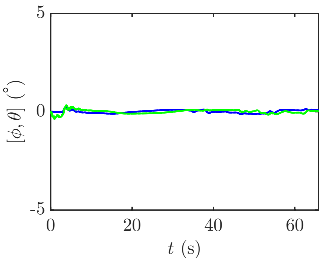
(b) Virtual image $\mathcal{V}$



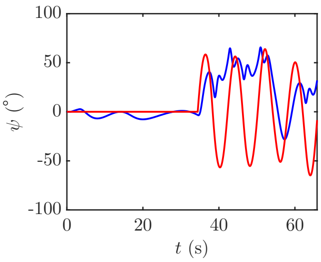
(c) Image features



(d) Lateral position

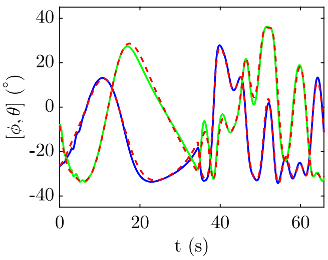


(e) Roll and pitch

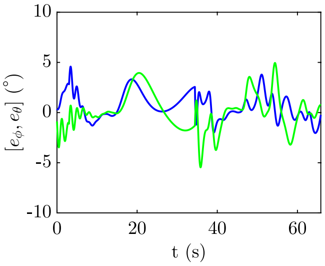


(f) Yaw ψ, ψ_t

Fig. 14. Simulation using a dynamic IBVS based on the virtual camera $\mathcal{V}$. The quadrotor (blue) follows a car (red).



(a) Roll and pitch estimates (solid) and actual values (dashed red)



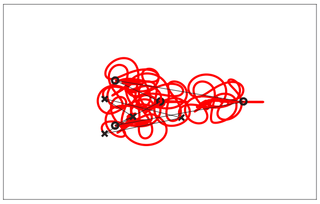
(b) Error in estimation

Fig. 15. Roll (blue) and pitch (green) estimation using a homography decomposition.

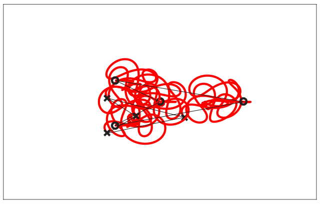
backward and forward projection error:

$$e_f = HY_1 - Y_2, \quad e_b = H^{-1}Y_2 - Y_1.$$

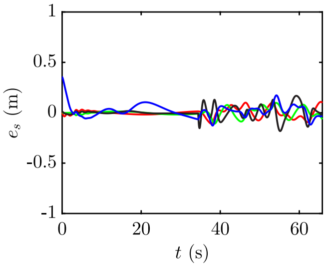
These errors are less than one pixel over the entire simulation.



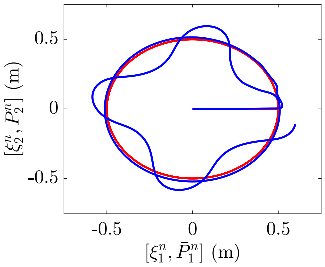
(a) Camera image



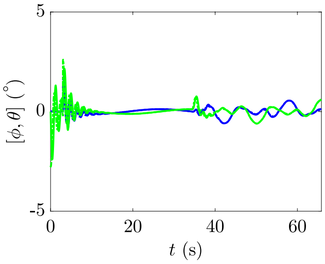
(b) Virtual image $\mathcal{V}$



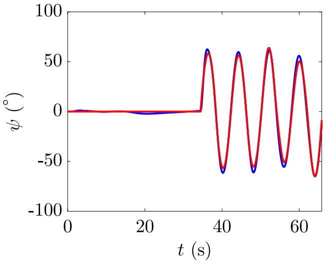
(c) Image features



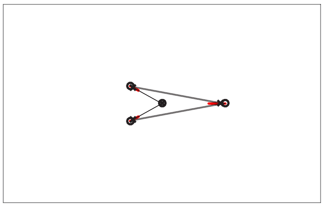
(d) Lateral position



(e) Roll and pitch



(f) Yaw ψ, ψ_t



(g) Virtual image $\mathcal{V}'$

Fig. 16. Simulation while using the proposed algorithm where the quadrotor (blue) followed a car (red).

7. Conclusion

In this work, we investigated a nonlinear dynamic IBVS for a quadrotor which regulates the relative position and yaw to a moving target. The control relies on four or more visual feature points in a plane. These point features are relatively easy to track, and the target geometry can be generically defined and is not assumed known. The general nature of the target makes the visual servoing law broadly applicable. The stability of the closed-loop is proven in the horizontal target case. Experimental results show the performance of the method in [23] for a horizontal target case. These results show robustness to unmodeled effects such as nonideal inner loop performance, mass imbalances causing a nondiagonal inertia matrix J, change in battery voltage causing a nonconstant thrust coefficient k_f, and relative pose between camera frame $\mathcal{C}$ and body frame $\mathcal{B}$. For the nonhorizontal moving target case, simulation results demonstrate the proposed algorithm's better performance relative to previously published work in [23] where the target was assumed horizontal and static.

Future work includes implementing the computer vision system on-board and extending the controller to remove the reliance on the Vicon system for linear velocity. Optical flow could provide this velocity. We observed experimentally that lateral position tracking error was degraded due to limited inner loop bandwidth. Future work can focus on improving inner loop response.

Appendix A. Homography Matrix and Decomposition

Following [31, 32], we consider two views of a point on a planar target as shown in Fig. A.1. The two camera frames $\mathcal{C}^1 = \{c_1^1, c_2^1, c_3^1\}$ and $\mathcal{C}^2 = \{c_1^2, c_2^2, c_3^2\}$ are related by a translation r^1 and rotation R. We denote the image plane coordinates of K points in the two cameras to be $y_i^1, y_i^2 \in \mathbb{R}^2, 1 \leq i \leq K$. The homogeneous representations of these vectors are denoted $Y_i^1, Y_i^2 \in \mathbb{R}^3, 1 \leq i \leq K$. A homography mapping $H \in \mathbb{R}^{3\times 3}$ is a matrix which transforms Y_i^1 to Y_i^2 according to

$$Y_i^2 \sim HY_i^1.$$

We note that the matrix H can be arbitrarily scaled, therefore it has eight degrees of freedom. When a homography is induced by a static plane it can be decomposed into a translation r^1 and a rotation R assuming that the optical center of the camera never passes through the plane. Such a situation is physically impossible as the camera would have

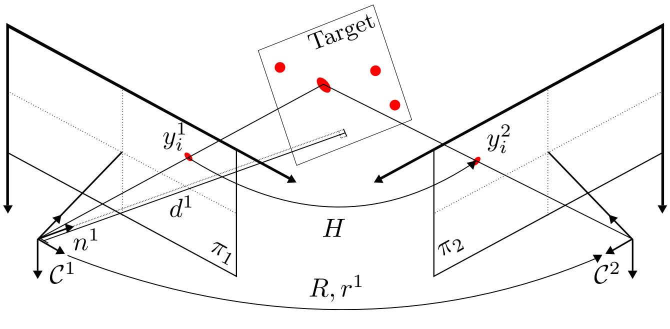


Fig. A.1. A Homography matrix transforms the image coordinates $Y_i^1 \in \mathbb{R}^3$ into $Y_i^2 \in \mathbb{R}^3$ of two views of the same 3D point.

to pass through the target. That is, we have

$$H = A\left(R + \frac{1}{d^1} r^1 {n^1}^T\right) A^{-1}, \tag{A.1}$$

where A is the matrix of intrinsic camera parameters, d^1 is the perpendicular distance from the camera frame $\mathcal{C}^1$ to the target plane, n^1 is the unit normal of the target plane expressed in $\mathcal{C}^1$, r^1 is the displacement of the origin of $\mathcal{C}^2$ relative to the origin of $\mathcal{C}^1$ expressed in $\mathcal{C}^1$, and R is the rotation matrix which describes the relative orientation of $\mathcal{C}^1$ and $\mathcal{C}^2$. Without loss of generality we take $A = I$ which assumes camera calibration has been performed. In practice, cameras can be easily calibrated to subpixel error. As an alternative, the points in the images can be normalized using a similarity transform that moves the centroid of the points in the images to the origin and scales the average distance of the points from the origin to $\sqrt{2}$. This approach yields acceptable results in practice. Normalization, via either the camera's intrinsic parameters or by a similarity transform, is an important step of the algorithm to provide suitable accuracy in the estimate of the homography [31].

The homography

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

can be calculated using at least $K \geq 4$ co-planar point correspondences in the two images using the planar homography constraint

$$\mathrm{sk}(Y_i^2) H Y_i^1 = 0, \quad 1 \leq i \leq K. \tag{A.2}$$

Since the constraint is linear we can rewrite it as

$$\chi H^s = 0, \tag{A.3}$$

where $H^s = [H_{11}, H_{21}, H_{31}, H_{12}, H_{22}, H_{32}, H_{13}, H_{23}, H_{33}]^T$,

$$\chi = \begin{bmatrix} \mathrm{kron}(Y_1^1, \mathrm{sk}(Y_1^2))^T \\ \mathrm{kron}(Y_2^1, \mathrm{sk}(Y_2^2))^T \\ \vdots \\ \mathrm{kron}(Y_K^1, \mathrm{sk}(Y_K^2))^T \end{bmatrix} \in \mathbb{R}^{3K\times 9} \tag{A.4}$$

and $\mathrm{kron}(A, B) \in \mathbb{R}^{mk\times nl}$ is the Kronecker product of $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{k\times l}$:

$$\mathrm{kron}(A, B) = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}. \tag{A.5}$$

To solve (A.3) uniquely (up to a scale factor) we need $\mathrm{rank}(\chi) = 8$. This is only possible if there exists a set of 4 points out of K on the plane such that no three of them are co-linear. Since Y_i^j contains noise we find the optimal χ by solving

$$\min_{||H^s||_2=1} ||\chi H^s||_2 \tag{A.6}$$

using linear least squares to obtain H^s up to a scale factor. To solve (A.6), we compute the SVD of $\chi = U\Sigma V^T$ where $U, V \in \mathbb{R}^{9\times 9}$ are orthogonal, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_9) \in \mathbb{R}^{9\times 9}, \sigma_1 \geq \cdots \geq \sigma_9 > 0$. Since V is orthogonal and defining $\xi = V^T H^s$ then $||H^s||_2 = ||VV^T H^s||_2 = ||V\xi||_2 = ||\xi||_2$. Hence,

$$\begin{aligned}\min_{||H^s||_2=1} ||\chi H^s||_2^2 &= \min_{||\xi||_2^2=1} ||U\Sigma\xi||_2^2 = \min_{||\xi||_2^2=1} ||\Sigma\xi||_2^2 \\ &= \min_{||\xi||_2^2=1} \sigma_1^2\xi_1^2 + \cdots + \sigma_9^2\xi_9^2 = \sigma_9^2.\end{aligned}$$

Thus the minimizing $\xi = [0, 0, \ldots, 0, 1]^T$ and

$$H_l^s = V[0, 0, \ldots, 0, 1]^T, \tag{A.7}$$

i.e., the ninth column of V. We form a scaled homography matrix H_l from solution H_l^s, i.e.,

$$H = \lambda H_l. \tag{A.8}$$

The magnitude of scale factor λ is given by

$$|\lambda| = \sigma_2(H_l), \tag{A.9}$$

where $\sigma_2(H_l)$ is the second largest singular value of H_l. To ensure the sign of H is correct we verify the constraint

$$(Y_i^2)^T H Y_i^1 > 0, \quad 1 \leq i \leq K. \tag{A.10}$$

A.1. *Homography decomposition*

There are multiple algorithms to decompose a homography matrix [31, 32, 39]. Here we will use a modified version of the SVD algorithm. There are four mathematical solutions to the decomposition of H into R, $\frac{1}{d^1}r^1$, and n^1. Only two of these solutions are physically meaningful, and to choose the correct solution additional information about the scene is required.

Given that $H^T H$ is symmetric we can diagonalize it as

$$H^T H = V\Sigma V^T, \tag{A.11}$$

where $V = [v_1\ v_2\ v_3] \in \mathrm{SO}(3), v_i \in \mathbb{R}^3$, and $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^3)$, where $\sigma_i, 1 \leq i \leq 3$ denote the singular values of the estimated H. Next, we define two unit length vectors u_1 and u_2 such that their length is preserved under the map H and that $\{v_2, u_1, \mathrm{sk}(v_2)u_1\}$ and $\{v_2, u_2, \mathrm{sk}(v_2)u_2\}$ are orthonormal bases of $\mathbb{R}^3$. We have the expressions

$$\begin{aligned} u_1 &= \frac{\sqrt{1-\sigma_3^2}v_1 + \sqrt{\sigma_1^2 - 1}v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \\ u_2 &= \frac{\sqrt{1-\sigma_3^2}v_1 - \sqrt{\sigma_1^2 - 1}v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \end{aligned} \tag{A.12}$$

We define the following matrices

$$\begin{aligned} U_1 &= [\,v_2 \quad u_1 \quad \mathrm{sk}(v_2)u_1\,] \\ U_2 &= [\,v_2 \quad u_2 \quad \mathrm{sk}(v_2)u_2\,] \\ W_1 &= [\,Hv_2 \quad Hu_1 \quad \mathrm{sk}(Hv_2)Hu_1\,] \\ W_2 &= [\,Hv_2 \quad Hu_2 \quad \mathrm{sk}(Hv_2)Hu_2\,] \end{aligned} \tag{A.13}$$

such that

$$\begin{aligned} RU_1 &= W_1 \\ RU_2 &= W_2 \end{aligned}$$

Due to the sign ambiguity in $\frac{1}{d^1}r^1 n^{1\,T}$ we obtain four decompositions of $H = R + \frac{1}{d^1}r^1 n^{1\,T}$ denoted $\{R_i, \frac{1}{d^1}r_i^1, n_i^1\}$, $i = 1, 2, 3, 4$ as shown in Table A.1. From these four solutions we can extract two relevant solutions using the positive depth constraint

$$(n^1)^T e_3 > 0 \tag{A.14}$$

to obtain two solutions.

The control method requires the scaled relative pose between the desired and current frames. We propose two methods for picking the correct solution from the homography decomposition data. The first method computes the homography between consecutive images and relies on the fact that images vary continuously in time. This implies the correct solution for R, n^1, and $\frac{1}{d^1}r^1$ are also continuous in time. The incorrect solution has no guarantee of continuity and can make large jumps. Initially we pick one of the solutions arbitrarily which we assume is correct. We monitor both solutions for discontinuities in the parameters. If one of the solutions exhibits a discontinuity, we can determine the correct solution. This method suffers from the

Table A.1. Four solutions of the decomposition of $H = R + \frac{1}{d^1}rn^{1\,T}$ into $\{R_i, \frac{1}{d^1}r_i^1, n_i^1\}$, $i = 1, 2, 3, 4$.

i	R_i	n_i^1	$\frac{1}{d^1}r_i^1$
1	$W_1U_1^T$	$\mathrm{sk}(v_2)u_1$	$(H - W_1U_1^T)\,\mathrm{sk}(v_2)u_1$
2	$W_2U_2^T$	$\mathrm{sk}(v_2)u_2$	$(H - W_2U_2^T)\,\mathrm{sk}(v_2)u_2$
3	R_1	$\mathrm{sk}(u_1)v_2$	$(W_1U_1^T - H)\,\mathrm{sk}(v_2)u_1$
4	$W_2U_2^T$	$\mathrm{sk}(u_2)v_2$	$(W_2U_2^T - H)\,\mathrm{sk}(v_2)u_2$

following limitations. First, it requires slow movement in the image and a fast frame rate. Second, the two solutions can meet making it impossible to determine the correct solution for subsequent time. In this case, the method is reset and we must arbitrarily choose a solution. The method relies on the robustness of the control law to incorrect choices of the solution.

In this work, we chose a simpler method to pick the correct solution which relies on a known target plane normal vector n^1 in the initial image. Rather than calculating the homography between consecutive frames we compute the homography between the initial and current images. This helps to eliminate any accumulation of error in the estimate of the scaled relative pose. The choice of correct solution is determined by comparing the estimated normal vector with its known value. Assuming a known target image and normal is not restrictive in that tracking algorithms require an initial image. We require additionally that the user specifies the initial target orientation. We remark that the method does not require any geometric information of the target and maintains the benefits of IBVS given that the control law is still calculated directly in the image plane.

References

[1] F. Kendoul, Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems, *J. Field Robotics* **29**(2) (2012) 315–378.

[2] S. Hutchinson, G. D. Hager and P. I. Corke, A tutorial on visual servo control, *IEEE Trans. Robot. Autom.* **12**(5) (1996) 651–670.

[3] O. Shakernia, Y. Ma, T. J. Koo and S. Sastry, Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control, *Asian J. Control* **1**(3) (1999) 128–145.

[4] E. Altug, J. Ostrowski and C. Taylor, Control of a quadrotor using dual camera visual feedback, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan (Sep. 2003), pp. 4294–4299.

[5] A. D. Wu, E. N. Johnson and A. A. Proctor, Vision-aided inertial navigation for flight control, *J. Aeros. Comp. Inf. Com.* **2**(9) (2005) 348–360.

[6] L. Mejias, P. Campoy, S. Saripalli and G. S. Sukhatme, A visual servoing approach for tracking features in urban areas using an autonomous helicopter, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Orlando, FL (May 2006), pp. 2503–2508.

[7] S. Azrad, F. Kendoul and K. Nonami, Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm, *J. Syst. Des. Dyn.* **4**(2) (2010) 255–268.

[8] L. Garcia Carrillo, E. Rondon, A. Sanchez, A. Dzul and R. Lozano, Stabilization and trajectory tracking of a quad-rotor using vision, *J. Intell. Robot. Syst.* **61**(1) (2011) 103–118.

[9] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen and M. Pollefeys, Vision-based autonomous mapping and exploration using a quadrotor MAV, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Portugal (Oct. 2012), pp. 4557–4564.

[10] F. Chaumette and S. Hutchinson, Visual servo control. II. Advanced approaches, *IEEE Robot. Autom. Mag.* **14**(1) (2007) 109–118.

[11] H. Jabbari Asl, G. Oriolo and H. Bolandi, An adaptive scheme for image-based visual servoing of an underactuated UAV, *Int. J. Robot. Autom.* **29**(1) (2014) 92–104.

[12] T. Hamel and R. Mahony, Visual servoing of an under-actuated dynamic rigid-body system: An image-based approach, *IEEE Trans. Robot. Autom.* **18**(2) (2002) 187–198.

[13] O. Bourquardez, R. Mahony, T. Hamel and F. Chaumette, Stability and performance of image based visual servo control using first order spherical image moments, in *Proc. IEEE/RSJ Int. Intelligent Robots and Systems Conf.*, Beijing, China (Oct. 2006), pp. 4304–4309.

[14] N. Guenard, T. Hamel and R. Mahony, A practical visual servo control for an unmanned aerial vehicle, *IEEE Trans. Robot. Autom.* **24**(2) (2008) 331–340.

[15] R. Mahony, P. Corke and T. Hamel, Dynamic image-based visual servo control using centroid and optic flow features, *J. Dyn. Syst.* **130**(1) (2007) 1–12.

[16] F. L. Bras, R. Mahony, T. Hamel and P. Binetti, Dynamic image-based visual servo control for an aerial robot: Theory and experiments, *Int. J. Optomechatronics* **2**(3) (2008) 296–325.

[17] N. Metni and T. Hamel, Visual tracking control of aerial robotic systems with adaptive depth estimation, *Int. J. Control Syst.* **5**(1) (2007) 51–60.

[18] H. de Plinval, P. Morin, P. Mouyon and T. Hamel, Visual servoing for underactuated VTOL UAVs: A linear, homography-based approach, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Shanghai, China (May 2011), pp. 3004–3010.

[19] H. de Plinval, P. Morin, P. Mouyon and T. Hamel, Visual servoing for underactuated VTOL UAVs: A linear, homography-based framework, *Int. J. Robust. Nonlinear Control* **24**(16) (2013) 2285–2308.

[20] R. Ozawa and F. Chaumette, Dynamic visual servoing with image moments for an unmanned aerial vehicle using a virtual spring approach, *Adv. Robotics* **27**(9) (2013) 683–696.

[21] D. Lee, T. Ryan and H. Kim, Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN (May 2012), pp. 971–976.

[22] D. Lee, H. Lim, H. Kim, Y. Kim and K. Seong, Adaptive image-based visual servoing for an underactuated quadrotor system, *J. Guid. Control. Dyn.* **35**(4) (2012) 1335–1353.

[23] G. Fink, H. Xie, A. F. Lynch and M. Jagersand, Nonlinear dynamic visual servoing of a quadrotor, *J. Unmanned Vehicle Syst.* **3**(1) (2015) 1–21.

[24] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel and L. Eck, Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle, *IEEE Trans. Robot. Autom.* **25**(3) (2009) 743–749.

[25] G. Fink, H. Xie, A. F. Lynch and M. Jagersand, Experimental validation of dynamic visual servoing for a quadrotor using a virtual camera, in *Proc. IEEE Int. Conf. Unmanned Aircraft Systems*, Denver, CO (June 2015), pp. 1231–1240.

[26] G. M. Hoffmann, H. Huang, S. L. Waslander and C. J. Tomlin, Precision flight control for a multi-vehicle quadrotor helicopter testbed, *Control Eng. Pract.* **19**(9) (2011) 1023–1036.

[27] S. Omari, M.-D. Hua, G. Ducard and T. Hamel, Nonlinear control of VTOL UAVs incorporating flapping dynamics, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Tokyo (Nov. 2013), pp. 2419–2425.

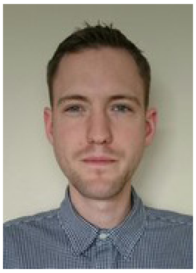
[28] S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, Ph.D. thesis, Ecole Polytechnique Federale de Luasanne (2007).

[29] R. Beard and T. McLain, *Small Unmanned Aircraft: Theory and Practice* (Princeton University Press, Princeton, NJ, 2012).

[30] M. Bangura, M. Melega, R. Naldi and R. Mahony, Aerodynamics of rotor blades for quadrotors, ArXiv e-print 1601.00733 (Jan. 2016).

[31] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Vol. 1, 2nd edn. (Cambridge University Press, Cambridge, England, 2003).

[32] Y. Ma, S. Soatto, J. Kosecka and S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models* (Springer, New York, NY, 2003).
[33] H. Xie and A. F. Lynch, State transformation-based dynamic visual servoing for an unmanned aerial vehicle, *Int J. Control* **89**(5) (2016) 892–908.
[34] F. Chaumette, Image moments: A general and useful set of features for visual servoing, *IEEE Trans. Robotics* **20**(4) (2004) 713–723.
[35] O. Tahri and F. Chaumette, Point-based and region-based image moments for visual servoing of planar objects, *IEEE Trans. Robotics* **21**(6) (2005) 1116–1127.
[36] H. Lim, J. Park, D. Lee and H. Kim, Build your own quadrotor: Open-source projects on unmanned aerial vehicles, *IEEE Robotics Autom. Maga.* **19**(3) (2012) 33–45.
[37] L. Meier, D. Honegger and M. Pollefeys, PX4: A node-based multi-threaded open source robotics framework for deeply embedded platforms, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Seattle, WA (May 2015), pp. 6235–6240.
[38] Pixy CMUcam5, Available at http://www.cmucam.org/projects/cmucam5/wiki/ [Accessed Sep. 1, 2016].
[39] O. Faugeras, Q.-T. Luong and T. Papadopoulou, *The Geometry of Multiple Images: The Laws that Govern the Formation of Images of a Scene and Some of their Applications* (MIT Press, Cambridge, MA, USA, 2001).



Geoff Fink received his B.Sc. in Computer Engineering in 2007 from the University of Alberta and M.Sc. in Robotics Engineering in 2011 from the University of Guadalajara. Since September 2012 he has been a Ph.D. student at the University of Alberta.

His current research interests include nonlinear control, unmanned aerial vehicles, and visual servoing.

Hui Xie received his B.Sc. degree in mechanical engineering from Harbin Engineering University in 2007, the M.Sc. degree in mechatronics engineering from Harbin Institute of Technology in 2009, and Ph.D. in Electrical and Computer Engineering from the University of Alberta in 2016. Currently, he is a research fellow in the School of Mechanical and Aerospace Engineering at Nanyang Technological University.

His research interests include nonlinear control theory, state estimation, and vision-based control with applications to unmanned aerial vehicles and mobile robots.

Alan F. Lynch obtained his B.A.Sc. at the University of Toronto in Engineering Science (Electrical Option) in 1991, M.A.Sc. in Electrical Engineering from University of British Columbia in 1994, and Ph.D. in Electrical and Computer Engineering from the University of Toronto in 1999. Since 2001 he has been a faculty member at the Department of Electrical & Computer Engineering, University of Alberta and currently holds the rank of Full Professor.

His interests include nonlinear control and its application to electrical and electromechanical systems including power converters, unmanned systems, and self-bearing motors.

Martin Jagersand studied physics at Chalmers Sweden (M.Sc. 1991). He was awarded a Fulbright fellowship for graduate studies in the United Sates. He studied Computer Science at the University of Rochester, NY (M.Sc. 1994, Ph.D. 1997). He held an NSF CISE postdoc fellowship at Yale University, and then was a research faculty in the Engineering Research Center for Surgical Systems and Technology at Johns Hopkins University. He is now a faculty member at the University of Alberta.

His research interests are in Computer Vision, Graphics and Robotics, especially visual servoing.