Robotica (2016) volume 34, pp. 2009–2026. © Cambridge University Press 2014
doi:10.1017/S0263574714002707

Dynamic IBVS of a rotary wing UAV using line features

Hui Xie†, Alan F. Lynch*† and Martin Jagersand‡

†*Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2V4*
‡*Department of Computing Science, University of Alberta, Edmonton, Alberta, T6G 2E8*

(Accepted October 30, 2014. First published online: December 9, 2014)

SUMMARY
In this paper we propose a *dynamic* image-based visual servoing (IBVS) control for a rotary wing unmanned aerial vehicle (UAV) which directly accounts for the vehicle's underactuated dynamic model. The motion control objective is to follow parallel lines and is motivated by power line inspection tasks where the UAV's relative position and orientation to the lines are controlled. The design is based on a virtual camera whose motion follows the onboard physical camera but which is constrained to point downwards independent of the vehicle's roll and pitch angles. A set of image features is proposed for the lines projected into the virtual camera frame. These features are chosen to simplify the interaction matrix which in turn leads to a simpler IBVS control design which is globally asymptotically stable. The proposed scheme is adaptive and therefore does not require depth estimation. Simulation results are presented to illustrate the performance of the proposed control and its robustness to calibration parameter error.

KEYWORDS: Unmanned Aerial Vehicle (UAV); Image-based visual servoing; Virtual camera; Adaptive nonlinear control.

1. Introduction

Small rotary wing Unmanned Aerial Vehicles (UAVs) such as ducted fans, quadrotors, and traditional helicopters have unique flight characteristics which make them suitable for applications such as search and rescue, infrastructure inspection, and surveillance. A survey of recent developments in rotary wing UAVs is in.[1] UAVs provide a number of benefits for inspection applications such as improved safety, reduced cost, and improved inspection data quality. There is particular interest from electrical utility companies in using small UAVs to inspect the infrastructure used for the transmission of electricity, e.g..[2,3] Electrical utilities are required to frequently inspect transmission lines which can span thousands of kilometers and adverse terrain. This inspection is required by law and is important to ensure the safety and reliability of the power system. A traditional solution to inspection involves costly, inefficient, and potentially dangerous manned flights. Given the magnitude and importance of the inspection problem, even small improvements in efficiency can lead to significant gain.

In this paper, we focus on the solution of a specific motion control problem for a small rotary wing UAV which can improve the inspection of electrical transmission lines. The vehicle is equipped with a single body-fixed downwards-facing camera. We use this video camera indirectly as a position sensor to provide feedback for motion control. The control is termed *visual servoing* since it depends on feedback provided by a camera. We take advantage of the approximately linear structure of the conductors which distinguish them from their background. We use their distinctive visual features to control the relative pose between the vehicle and two or more power lines. In particular, we regulate the lateral displacement, yaw, and height relative to the lines. As well, the velocity of the vehicle is controlled in the direction of the lines. Such a control problem is of practical interest when performing inspection of power line infrastructure since it provides a consistent view of the lines so

* Corresponding author. Email: alan.lynch@ualberta.ca

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

as to optimize the quality of the inspection. As mentioned in work,[4] where the problem of visual servoing for guidance along curved paths is considered, any smooth curve with limited curvature can be locally approximated by a sequence of tangent lines. Furthermore, when viewed from above power lines have relatively low curvature compared to other infrastructure such as roads. These two factors justify restricting our attention to the simplified case of parallel lines.

In[5,6] visual servoing was categorized into two basic approaches: *position-based visual servoing* (PBVS) and *image-based visual servoing* (IBVS). Although there exist PBVS work for UAVs to follow linear structures, (e.g.[7,8] for road following), we choose to focus on IBVS methods in this paper. In IBVS the error signal is constructed directly from image features, and this eliminates the need for position reconstruction which requires camera calibration and a prior 3D model of the target. In general, IBVS is relatively insensitive to camera calibration error.[9] Since for IBVS the task function of visual servoing is constructed in the image space, the target object tends not to leave the field of view.

It is pointed out in[6] that the dynamics of the robot has to be considered for high speed tasks or when the controlled system is underactuated. Conventional visual servoing is usually based on kinematic models. This effectively assumes that the robot can perfectly follow a given velocity command. Such approaches have a two loop control structure: the inner loop accepts a reference velocity and controls the robot's end effector velocity with a relatively high bandwidth. An outer loop uses video camera feedback and operates at a relatively low frequency due to the low rate at which images are acquired and processed.[10] Recently, developments in high speed cameras and image processing techniques make high sampling rates possible. For example, the HiSpec 5 camera from Fastec Imaging Corporation can provide high resolution images of 1696×1710 pixels at up to 523 FPS, and the QVGA-Size Pixel-Parallel Image Processor in[11] can achieve frame rates as high as 1000 FPS. In the case of a fast camera, accounting for the dynamics of the robot becomes particularly relevant in improving closed-loop motion control performance. A main challenge of applying IBVS to UAVs comes from the underactuated nature of their dynamics. The rotational and translational degrees of freedom are coupled and cannot be controlled independently. A typical IBVS solution is to choose image features with passivity properties. In this solution, the control design is simplified since the time derivative of the norm of the image feature error depends only on 3D translational velocity and yaw rate. Popular choices for point features are image moments computed in a spherically projected image plane.[12] However, this feature is insensitive to motion in altitude due to an ill-conditioned interaction matrix. To overcome this, various modified spherical image moments features have been proposed.[13] A similar approach to[12] for linear features is considered in.[14] Here, the line image moment with bi-normalized Plücker coordinates is presented, but the ill-conditioning of the interaction matrix is not addressed. Besides spherical image moments of point features, another feature can be obtained from the homography matrix between the desired and current images. This matrix embeds both translation and orientation information of the vehicle.[15–17] However, those approaches either control partial motion of the UAV or are limited to a small range of image feature error. A virtual spring approach is proposed in[18,19] to deal with scenarios where the chosen image feature kinematics do not have passivity properties. This approach assumes the image and target planes remain parallel. This assumption is restrictive since lateral UAV motion leads to an image plane which is not parallel to the target plane. A different approach is in[20] which is based on a so-called virtual image plane. This plane has zero roll and pitch angle and has the same position and yaw angle of the real camera frame. The features are extracted in the real camera image and transformed to the virtual camera image. To implement a virtual camera we require the transformation formula which is derived from geometry. The transformation depends on estimates of the roll-pitch angles of the camera. These angle measurements are commonly obtained from an Attitude and Heading Reference System (AHRS) which is integrated into most UAV autopilots. Instead of a virtual camera, we can use a pan-tilt or gimballed mechanism to ensure a real camera has zero roll and pitch. On one hand, a camera with a gimbal simplifies the control design as no transformation to a virtual frame is required. This implies that intrinsic camera parameters, which are required by the virtual camera transformation, would not be required for IBVS. On the other hand, a camera with gimbal adds cost, complexity, and weight to the platform which may not be suitable in some applications. The gimbal system will have its own dynamics to consider while the virtual camera uses an algebraic computation. High bandwidth for the gimbal system requires a costly servosystem. The use of software to implement a virtual camera is analogous to computing a spherical projection in software instead of using spherical camera hardware.[12]

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

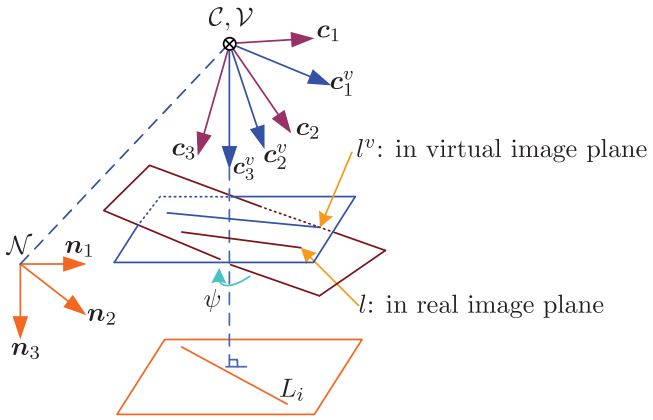


Fig. 1. Frame definitions.

In this paper we adopt the virtual camera approach in[20] due to its simple interaction matrix. We provide a method for converting projected lines in the real camera image to lines in the virtual camera image. A set of line features in the virtual image is proposed in order to yield a simple decoupled interaction matrix. Based on these features we develop an adaptive dynamic IBVS controller which is proven to be globally asymptotically stable. The control does not require target depth information which is often assumed in classical IBVS approaches. The proposed approach requires a measurement of linear velocity which can be obtained from GPS. Attitude and angular velocity measurements are also required and can be obtained from the AHRS.

The paper is organized as follows. The UAV dynamics and line projection model are given in Section 2. Then Section 3 provides the transformation method from real image lines to virtual camera lines. The feature selection and its interaction matrix are given in Section 4. Section 5 presents the proposed IBVS controller and simulation results are given in Section 6. Finally, concluding remarks are provided in Section 7.

2. Modelling

In this section we first introduce several reference frames for the UAV dynamics and line projection model. We specify a rigid body UAV dynamics which is used to derive the dynamic IBVS control law. Finally, we present a parametrization of the projected line in the image and derive its kinematics using a similar approach to.[21]

2.1. Frames definition

As shown in Fig. 1, three reference frames are introduced. A navigation frame $\mathcal{N}$ is fixed to some point on the earth with a basis $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$, which are oriented north, east, and down, respectively. The frame $\mathcal{N}$ is approximately inertial as the earth rotates slowly. We assume the camera is fixed at the centre of mass (CoM) of the UAV, and define a camera frame $\mathcal{C}$ with basis $\{\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3\}$ oriented forward, right, down, respectively. Next, we define a virtual camera frame $\mathcal{V}$ with basis $\{\boldsymbol{c}_1^v, \boldsymbol{c}_2^v, \boldsymbol{c}_3^v\}$ which is attached to the real camera but with zero roll and pitch angle. That is, the virtual camera's image plane is always parallel to the $\boldsymbol{n}_1, \boldsymbol{n}_2$ plane. The relative orientation of the reference frames can be described by a rotation matrix $\boldsymbol{R} \in SO(3)$. In this paper we parameterize $\boldsymbol{R}$ with the so-called roll-pitch-yaw (ϕ-θ-ψ) or ZYX Euler angles. The orientation of $\mathcal{N}$ relative to $\mathcal{C}$ is given by

$$\boldsymbol{R}_C^N = \boldsymbol{R}_{\boldsymbol{n}_3}(\psi)\boldsymbol{R}_{\boldsymbol{n}_2}(\theta)\boldsymbol{R}_{\boldsymbol{n}_1}(\phi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

where $c_\xi = \cos\xi$ and $s_\xi = \sin\xi$.

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

2.2. Rotary wing UAV dynamics

We write the translational dynamics in virtual camera frame $\mathcal{V}$ and the rotational dynamics in camera frame $\mathcal{C}$ as

$$\dot{\boldsymbol{v}}^v = -\boldsymbol{S}(\dot{\psi}\boldsymbol{E}_3)\boldsymbol{v}^v + \frac{\boldsymbol{F}^v}{m} \tag{1a}$$

$$\dot{\boldsymbol{R}}_C^N = \boldsymbol{R}_C^N \boldsymbol{S}(\boldsymbol{\omega}^c) \tag{1b}$$

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

where $\boldsymbol{v}^v$ is the UAV's translational velocity expressed in $\mathcal{V}$, $\boldsymbol{E}_3 = [0\ 0\ 1]^T$, $\boldsymbol{S}(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$, $\boldsymbol{\omega}^c$ is the angular velocity of the UAV expressed in $\mathcal{C}$, m and $\boldsymbol{J}$ are mass and inertia of the UAV, $\boldsymbol{\tau}^c$ is the external torque expressed in $\mathcal{C}$, and $\boldsymbol{F}^v$ is the external force expressed in $\mathcal{V}$. We consider a commonly used external force approximation for a rotary wing UAV which ignores higher order effects such as small body forces found in traditional helicopters,[22] i.e., the expression for $\boldsymbol{F}^v$ is approximated by

$$\boldsymbol{F}^v = -\boldsymbol{R}_{\theta\phi} T \boldsymbol{E}_3 + mg\boldsymbol{E}_3 \tag{2}$$

where

$$\boldsymbol{R}_{\theta\phi} = \boldsymbol{R}_{\boldsymbol{n}_2}(\theta)\boldsymbol{R}_{\boldsymbol{n}_1}(\phi) = \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi & -s_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi\ c_\theta \end{bmatrix}$$

the thrust is denoted T which creates a force along the $\boldsymbol{c}_3$ axis, and g is the gravity constant.

2.3. Single line features and the interaction matrix

Any straight line denoted as L expressed in $\mathcal{C}$ is the intersection of two planes

$$\begin{cases} \boldsymbol{h}_1^T \boldsymbol{P} + D_1 = A_1 X + B_1 Y + C_1 Z + D_1 = 0 \\ \boldsymbol{h}_2^T \boldsymbol{P} + D_2 = A_2 X + B_2 Y + C_2 Z + D_2 = 0 \end{cases} \tag{3}$$

where $\boldsymbol{h}_i = [A_i\ B_i\ C_i]^T$ is the normal vector of ith plane expressed in $\mathcal{C}$, $\boldsymbol{P} = [X\ Y\ Z]^T$ is the coordinate of any point P on L expressed in $\mathcal{C}$. The projection of P on the image plane is a 2D point feature p with homogeneous normalized coordinates $\bar{\boldsymbol{p}} = [m_1\ m_2\ 1]^T = [\frac{X}{Z}\ \frac{Y}{Z}\ 1]^T$. As in,[23] the corresponding coordinates in pixels are denoted $\tilde{\boldsymbol{p}} = [u\ v\ 1]^T$ with

$$\tilde{\boldsymbol{p}} = \boldsymbol{A}(f, u_0, v_0)\bar{\boldsymbol{p}}$$

where

$$\boldsymbol{A}(f, u_0, v_0) = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

the focal length of the pinhole camera in pixels is denoted f, and u_0, v_0 are the pixel coordinates of the camera's principal point. In this paper we assume u_0 and v_0 are known. Hence, by defining $y_1 = u - u_0$ and $y_2 = v - v_0$, a new parametrization of p is $\boldsymbol{p} = [y_1\ y_2\ 1]^T$ and we have kinematics relationship

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -\frac{f}{Z} & 0 & \frac{y_1}{Z} & \frac{y_1 y_2}{f} & -(f + \frac{y_1^2}{f}) & y_2 \\ 0 & -\frac{f}{Z} & \frac{y_2}{Z} & f + \frac{y_2^2}{f} & -\frac{y_1 y_2}{f} & -y_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^c \\ \boldsymbol{\omega}^c \end{bmatrix} \tag{4}$$

where $\boldsymbol{v}^c$ and $\boldsymbol{\omega}^c$ are translational and rotational velocities of the camera expressed in $\mathcal{C}$, respectively.

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

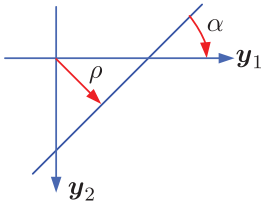


Fig. 2. A graph of a line in the image plane and its dependence on feature parameters α and ρ. The image plane basis vectors are denoted $\{\boldsymbol{y}_1, \boldsymbol{y}_2\}$.

If the degenerate case ($D_1 = D_2 = 0$) is excluded, which means that the origin of $\mathcal{C}$ does not lie on L, then the projected line denoted l in the image plane can be parametrized by $\boldsymbol{l}_I$ with following equation[21]

$$\boldsymbol{l}_I^T \bar{\boldsymbol{p}} = 0$$

with

$$\boldsymbol{l}_I = \begin{bmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 \end{bmatrix} \begin{bmatrix} D_2 \\ -D_1 \end{bmatrix}$$

It can be shown that the parametrization of the projected line l in pixels is

$$\boldsymbol{l}^T \tilde{\boldsymbol{p}} = 0$$

with

$$\boldsymbol{l} = \boldsymbol{B}(f, u_0, v_0)\boldsymbol{l}_I \tag{5}$$

and

$$\boldsymbol{B}(f, u_0, v_0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -u_0 & -v_0 & f \end{bmatrix}$$

Since the parametrization $\boldsymbol{l}$ is not minimal (e.g. $\boldsymbol{l}$ and $\sigma\boldsymbol{l}$ represent the same line where σ is some nonzero constant), a new pair of variables ρ and α are used to parametrize the projected line l as

$$\boldsymbol{l}_f^T \boldsymbol{p} = y_1 \sin\alpha + y_2 \cos\alpha - \rho = 0 \tag{6}$$

where $\boldsymbol{l}_f = [\sin\alpha \;\; \cos\alpha \;\; -\rho]^T$. This representation is ambiguous since the same line can be parametrized by $(\rho, \alpha + 2k\pi)$ and $(-\rho, \alpha + (2k+1)\pi)$. Hence, without loss of generality we restrict α to lie in $(-\frac{\pi}{2}, \frac{\pi}{2}]$. The parameter α represents the angle the line makes with the positive y_1-axis with positive α taken in the clockwise direction and restricted to the range $(-\frac{\pi}{2}, \frac{\pi}{2}]$. The parameter ρ represents a perpendicular distance of the line to the origin of the image (or principle point). The sign of ρ can be determined from (6). Figure 2 shows the graphical representation α and ρ. Differentiating (6) with respect to time yields

$$\dot{\rho} - (y_1 \cos\alpha - y_2 \sin\alpha)\dot{\alpha} = \dot{y}_1 \sin\alpha + \dot{y}_2 \cos\alpha \tag{7}$$

From (3), we can obtain

$$\frac{1}{Z} = -\frac{A_i y_1 + B_i y_2 + f C_i}{f D_i} \tag{8}$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

From (6), y_2 is expressed as a function of y_1 when $\cos\alpha \neq 0$. Alternatively, y_1 can be expressed as a function of y_2 when $\sin\alpha \neq 0$. By using (4) and (8), (7) is written

$$-\dot{\alpha} y_1 + \cos\alpha(\dot{\rho} + \dot{\alpha}\rho\tan\alpha) = y_1 \boldsymbol{K}_1 \cos\alpha \mathbf{v}^c + \boldsymbol{K}_2 \cos\alpha \mathbf{v}^c$$

where

$$\boldsymbol{K}_1 = \begin{bmatrix} \lambda_1 f \sin\alpha & \lambda_1 f \cos\alpha & -\lambda_1\rho & -\dfrac{\rho\tan\alpha}{f} & -\dfrac{\rho}{f} & -\dfrac{1}{\cos\alpha} \end{bmatrix}$$

$$\boldsymbol{K}_2 = \begin{bmatrix} \lambda_2 f \sin\alpha & \lambda_2 f \cos\alpha & -\lambda_2\rho & f\cos\alpha + \dfrac{\rho^2}{f\cos\alpha} & -f\sin\alpha & \rho\tan\alpha \end{bmatrix}$$

with $\lambda_1 = \frac{A_i - B_i\tan\alpha}{D_i f}$, $\lambda_2 = \frac{B_i\rho/\cos\alpha + f C_i}{D_i f}$, and $\mathbf{v}^c = [(\boldsymbol{v}^c)^T \ (\boldsymbol{\omega}^c)^T]^T$. This leads to

$$\begin{aligned} \dot{\alpha} &= -\boldsymbol{K}_1 \cos\alpha \mathbf{v}^c = \boldsymbol{L}_\alpha \mathbf{v}^c \\ \dot{\rho} &= (\boldsymbol{K}_2 + \rho\sin\alpha \boldsymbol{K}_1)\mathbf{v}^c = \boldsymbol{L}_\rho \mathbf{v}^c \end{aligned} \tag{9}$$

where

$$\boldsymbol{L}_\alpha = \begin{bmatrix} \lambda_\alpha f \sin\alpha & \lambda_\alpha f \cos\alpha & -\lambda_\alpha\rho & \dfrac{\rho\sin\alpha}{f} & \dfrac{\rho\cos\alpha}{f} & 1 \end{bmatrix}$$

$$\boldsymbol{L}_\rho = \begin{bmatrix} \lambda_\rho f \sin\alpha & \lambda_\rho f \cos\alpha & -\lambda_\rho\rho & \left(f + \dfrac{\rho^2}{f}\right)\cos\alpha & -\left(f + \dfrac{\rho^2}{f}\right)\sin\alpha & 0 \end{bmatrix}$$

with $\lambda_\alpha = -\frac{A_i\cos\alpha - B_i\sin\alpha}{D_i f}$, $\lambda_\rho = \frac{A_i\rho\sin\alpha + B_i\rho\cos\alpha + C_i f}{D_i f}$.

3. Converting Line Features from the Real Camera to the Virtual Camera

The proposed control relies on the notion of a virtual camera which shares the position and yaw of the real camera. The virtual camera is not implemented in hardware (e.g. using a gimbal system) but uses a conversion formula which depends on camera roll and pitch measurements which are provided by an AHRS. This section gives a geometric derivation of the formula for converting projected lines on the real camera image into projected lines on the virtual camera image.

Since the rotation matrix $\boldsymbol{R}_C^N$ maps a vector represented in $\mathcal{C}$ to $\mathcal{N}$, any vector $\boldsymbol{X}$ in the real camera frame is expressed as

$$\boldsymbol{X}^c = (\boldsymbol{R}_C^N)^T \boldsymbol{X}^n \tag{10}$$

And the same vector expressed in the virtual camera frame is

$$\boldsymbol{X}^v = (\boldsymbol{R}_{\boldsymbol{n}_3})^T \boldsymbol{X}^n \tag{11}$$

Based on (10), and (11) we can obtain

$$\boldsymbol{X}^v = \boldsymbol{R}_{\boldsymbol{n}_2}(\theta)\boldsymbol{R}_{\boldsymbol{n}_1}(\phi)\boldsymbol{X}^c = \boldsymbol{R}_{\theta\phi}\boldsymbol{X}^c$$

The line defined in (3) expressed in the virtual camera frame is

$$(\boldsymbol{h}_i^v)^T \boldsymbol{P}^v + D_i^v = A_i^v X^v + B_i^v Y^v + C_i^v Z^v + D_i^v = 0 \tag{12}$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

where $i = 1, 2$, $\boldsymbol{h}_i^v = [A_i^v \ B_i^v \ C_i^v]^T$ is the normal vector of ith plane expressed in $\mathcal{V}$. Based on (11) we have

$$\begin{aligned} 0 &= (\boldsymbol{R}_{\theta\phi}\boldsymbol{h}_i)^T \boldsymbol{P}^v + D_i^v \\ &= (\boldsymbol{h}_i)^T (\boldsymbol{R}_{\theta\phi})^T \boldsymbol{P}^v + D_i^v \\ &= (\boldsymbol{h}_i)^T \boldsymbol{P} + D_i^v = (\boldsymbol{h}_i)^T \boldsymbol{P} + D_i \end{aligned}$$

and deduce that $D_i = D_i^v$. Hence, the projected line l^v in the image plane of virtual camera frame $\mathcal{V}$ is parametrized as

$$(\boldsymbol{l}_I^v)^T \boldsymbol{b}^v = \begin{bmatrix} a^v & b^v & c^v \end{bmatrix} \begin{bmatrix} m_1^v \\ m_2^v \\ 1 \end{bmatrix} = 0$$

with

$$\boldsymbol{l}_I^v = \begin{bmatrix} \boldsymbol{h}_1^v & \boldsymbol{h}_2^v \end{bmatrix} \begin{bmatrix} D_2 \\ -D_1 \end{bmatrix} = \boldsymbol{R}_{\theta\phi} \begin{bmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 \end{bmatrix} \begin{bmatrix} D_2 \\ -D_1 \end{bmatrix} = \boldsymbol{R}_{\theta\phi} \boldsymbol{l}_I \tag{13}$$

From (5) and (13), we can obtain the parametrization of l^v as

$$\begin{aligned} \boldsymbol{l}^v &= \boldsymbol{B}(f, u_0, v_0)\boldsymbol{l}_I^v = \boldsymbol{B}(f, u_0, v_0)\boldsymbol{R}_{\theta\phi}\boldsymbol{l}_I \\ &= \boldsymbol{B}(f, u_0, v_0)\boldsymbol{R}_{\theta\phi}\boldsymbol{B}^{-1}(f, u_0, v_0)\boldsymbol{B}(f, u_0, v_0)\boldsymbol{l}_I \\ &= \boldsymbol{B}(f, u_0, v_0)\boldsymbol{R}_{\theta\phi}\boldsymbol{B}^{-1}(f, u_0, v_0)\boldsymbol{l} \end{aligned}$$

We denote an estimate of the camera's focal length as $\hat{f}$, then the line L projected in the virtual camera image plane can be parameterized as

$$\boldsymbol{l}^v = \boldsymbol{B}(\hat{f}, u_0, v_0)\boldsymbol{R}_{\theta\phi}\boldsymbol{B}^{-1}(\hat{f}, u_0, v_0)\boldsymbol{l}$$

In Section 6 we show by simulation that the proposed control is robust to a practically reasonable amount of error in the estimate of focal length. This is likely due to the generally accepted fact that IBVS is robust to intrinsic calibration error.[9] A mathematical analysis of this robustness property does not appear to be available in the literature and provides a future direction for research.

4. Feature Selection and Interaction Matrix

In this section, we select a set of image features from lines in the virtual camera image. Based on previous sections, the interaction matrix of the selected features is also derived.

As in (6) we parametrize the projected line l^v on virtual camera image plane as $\boldsymbol{l}_f^v = [\sin\alpha^v \ \cos\alpha^v \ -\rho^v]^T$

$$y_1^v \sin\alpha^v + y_2^v \cos\alpha^v - \rho^v = 0$$

Since our research focusses on power lines we make the following assumption

Assumption 1. $N > 1$ lines are parallel to each other and lie in a horizontal plane which is parallel to the $\boldsymbol{n}_1 - \boldsymbol{n}_2$ plane.

This assumption means that one of the planes in (12) can be chosen parallel to the image plane, i.e., $A_i^v = B_i^v = 0$. The ith line l_i^v is parametrized by ρ_i^v and α_i^v. From (9), the kinematics of α^v and ρ^v

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

can be written as

$$\dot{\alpha}_i^v = \boldsymbol{L}_\theta^T \begin{bmatrix} \boldsymbol{v}^v \\ \boldsymbol{\omega}^v \end{bmatrix} = \dot{\psi} \tag{14a}$$

$$\dot{\rho}_i^v = \boldsymbol{L}_\rho^T \begin{bmatrix} \boldsymbol{v}^v \\ \boldsymbol{\omega}^v \end{bmatrix} = -\frac{1}{Z^v} \begin{bmatrix} f \sin\alpha_i^v & f\cos\alpha_i^v & -\rho_i^v \end{bmatrix} \boldsymbol{v}^v \tag{14b}$$

where $\boldsymbol{v}^v$ is the translational velocity of the virtual camera expressed in $\mathcal{V}$, and Z^v is the unknown depth. From Assumption 1 it is straightforward to deduce that all α_i^v have the same value which is denoted α^v.

Inspired by the image moment based work in,[24] we propose a moment-like line features for IBVS. The line centroid ρ_g^v is defined as

$$\rho_g^v = \frac{1}{N}\sum_{i=1}^{N} \rho_i^v$$

Based on (14), we obtain

$$\dot{\rho}_g^v = -\frac{1}{Z^v} \begin{bmatrix} f \sin\alpha^v & f\cos\alpha^v & -\rho_g^v \end{bmatrix} \boldsymbol{v}^v \tag{15}$$

Next, the central moment μ_{20}^v is defined as

$$\mu_{20}^v = \sum_{i=1}^{N} \left(\rho_i^v - \rho_g^v\right)^2$$

Differentiating μ_{20}^v we get

$$\dot{\mu}_{20}^v = 2\sum_{i=1}^{N} \left(\rho_i^v - \rho_g^v\right)\left(\dot{\rho}_i^v - \dot{\rho}_g^v\right) = \frac{2\mu_{20}^v}{Z^v} v_3^v \tag{16}$$

Using (16), it can be easily obtained that

$$\frac{d}{dt}(Z^v\sqrt{\mu_{20}}) = 0$$

Hence, $Z^v\sqrt{\mu_{20}}$ is a constant value, namely

$$Z^v\sqrt{\mu_{20}} = Z^{v*}\sqrt{\mu_{20}^{v*}} \tag{17}$$

where Z^{v*} and μ_{20}^{v*} are desired constant values of Z^v and μ_{20}, respectively. By defining

$$a_n^v = \sqrt{\frac{\mu_{20}^{v*}}{\mu_{20}^v}}$$

and differentiating it we obtain

$$\dot{a}_n^v = -\frac{1}{Z^{v*}} \frac{Z^{v*}\sqrt{\mu_{20}^{v*}}}{Z^v\sqrt{\mu_{20}^v}} v_3^v = -\frac{1}{Z^{v*}} v_3^v \tag{18}$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

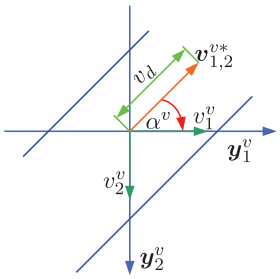


Fig. 3. The relationship (21) between $\boldsymbol{v}^{v*}$, α^v, and v_d. The vector $\boldsymbol{v}_{1,2}^{v*}$ is the projection of $\boldsymbol{v}^{v*}$ onto the image plane. The image plane basis vectors for the virtual camera are denoted $\{\boldsymbol{y}_1^v, \boldsymbol{y}_2^v\}$.

Finally, the image features used for controlling the motion of the UAV are defined as

$$s_1 = a_n^v \tag{19a}$$

$$s_2 = \beta \rho_g^v a_n^v \tag{19b}$$

$$s_3 = \sum_{1=1}^{N} \alpha_i^v / N \tag{19c}$$

The parameter β was included in the image feature definition to scale the range of s_2. This is to avoid the large value that would result from ρ_g^v which can take values in the range of a few hundred pixels. We choose a value for β based on the initial measured value of ρ_g^v, i.e., $\beta \approx 1/\rho_g^v(0)$. As mentioned above $\alpha_i^v = \alpha^v = s_3$. However, in practice we allow different values of α_i^v due to image noise.

Based on (15), (17), and (18) we can obtain

$$\dot{s}_1 = -\frac{1}{Z^{v*}} v_3^v \tag{20a}$$

$$\dot{s}_2 = -\frac{\beta f}{Z^{v*}} \begin{bmatrix} \sin\alpha^v & \cos\alpha^v & 0 \end{bmatrix} \boldsymbol{v}^v \tag{20b}$$

$$\dot{s}_3 = \dot{\psi} \tag{20c}$$

It can be seen from (20) that the interaction matrix has a simple decoupled structure. That is, the feature s_1 is only sensitive to velocity in the $\boldsymbol{c}_3^v$ direction. The feature s_2 is sensitive to the motion in the lateral direction to the lines. Clearly, the feature s_3 is only sensitive to the yaw rotation of the UAV. The simple dependence leads to a simpler derivation of the control law presented in the next section.

5. Dynamic IBVS

In this section, based on the selected image features given in the previous section, we first derive the image error kinematics and then propose a translational motion controller using the backstepping approach. We also give a simple controller for yaw motion.

5.1. Translational motion control

The motion control objective is to stabilize the UAV above parallel lines at a desired height. This height is determined by μ_{20}^{v*}, and when $\mu_{20}^v = \mu_{20}^{v*}$ the desired value $s_1^* = 1$. We also want to position the UAV laterally relative to the parallel lines. This means the desired value of $s_2^* = 0$. The value of s_3^* is 0 which implies translational motion can only occur along the "forward" direction of the lines. The "forward" direction of the line is defined by projecting the body $\boldsymbol{c}_1^v$-axis on the line. Clearly we avoid the singular case initial conditions where $\boldsymbol{c}_1^v$ is perpendicular to the line i.e., $\alpha(0) = \pi/2$. Finally, we control the velocity along the parallel lines to v_d. That is, as shown in Fig. 3, the desired

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

UAV velocity v^{v*} projected onto the direction of the line should be v_d. This means

$$(\boldsymbol{v}^{v*})^T\boldsymbol{\eta} - v_d = \begin{bmatrix} v_1^{v*} & v_2^{v*} & v_3^{v*} \end{bmatrix} \begin{bmatrix} \cos\alpha^v \\ -\sin\alpha^v \\ 0 \end{bmatrix} - v_d = 0 \tag{21}$$

The image feature error $\boldsymbol{e} = [\boldsymbol{e}_{yz}, e_\alpha]^T$ is

$$\boldsymbol{e}_{yz} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} s_1 - s_1^* \\ s_2 \end{bmatrix} \tag{22a}$$

$$e_\alpha = s_3 \tag{22b}$$

Based on (14a), (18), (19), (20), and (22), the kinematics of the image feature error is

$$\dot{\boldsymbol{e}}_{yz} = -\frac{1}{Z^{v*}}\boldsymbol{D}(\beta, f)\boldsymbol{L}_{yz}(\alpha^v)\boldsymbol{v}^v \tag{23a}$$

$$\dot{e}_\alpha = \dot{\psi} \tag{23b}$$

where the interaction matrix is

$$\boldsymbol{L}_{yz}(\alpha^v) = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha^v & \cos\alpha^v & 0 \end{bmatrix}, \text{ and } \boldsymbol{D}(\beta, f) = \begin{bmatrix} 1 & 0 \\ 0 & \beta f \end{bmatrix}$$

We develop a controller which is adaptive to two parameters γ_1 and γ_2:

$$\gamma_1 = \frac{1}{Z^{v*}} \tag{24a}$$

$$\gamma_2 = \frac{f}{\hat{f}Z^{v*}} \tag{24b}$$

Introducing these parameters eliminates the requirement of knowing the desired depth Z^{v*}. Also, three variables are introduced

$$\boldsymbol{q}_1 = \begin{bmatrix} \boldsymbol{e}_{yz} \\ v_d \end{bmatrix} \tag{25a}$$

$$\boldsymbol{q}_2 = \boldsymbol{Q}^{-1}\boldsymbol{H}(\beta, \hat{f})\boldsymbol{W}\boldsymbol{v}^v - \boldsymbol{q}_1 \tag{25b}$$

$$\boldsymbol{q}_3 = \boldsymbol{F}' + k_2\boldsymbol{q}_2 + \hat{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 \tag{25c}$$

where

$$\boldsymbol{H}(\beta, \hat{f}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \beta\hat{f} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{Q} = \begin{bmatrix} k_1\boldsymbol{I}_2 & \boldsymbol{0}_{2\times 1} \\ \boldsymbol{0}_{1\times 2} & 1 \end{bmatrix}, \boldsymbol{W}(\alpha^v) = \begin{bmatrix} \boldsymbol{L}_{yz}(\alpha^v) \\ \boldsymbol{\eta}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha^v & \cos\alpha^v & 0 \\ \cos\alpha^v & -\sin\alpha^v & 0 \end{bmatrix}$$

$$\boldsymbol{F}' = \boldsymbol{\Delta}_1\boldsymbol{W}\boldsymbol{F}^v, \ \boldsymbol{\Delta}_1 = \frac{\boldsymbol{Q}^{-1}\boldsymbol{H}(\beta, \hat{f})}{m}, \ \hat{\boldsymbol{M}} = \begin{bmatrix} \hat{\gamma}_1 & 0 & 0 \\ 0 & \hat{\gamma}_2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

k_1, k_2 are constant gains, and $\hat{\gamma}_1, \hat{\gamma}_2$ denote estimated values of γ_1, γ_2, respectively.

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

THEOREM 5.1. *If the control T and $\boldsymbol{\omega}^c$ satisfy*

$$\boldsymbol{\Delta}_1 \boldsymbol{W} \boldsymbol{R}_{\theta\phi} \left(\boldsymbol{S}(\boldsymbol{\omega}^c) T \boldsymbol{E}_3 + \dot{T} \boldsymbol{E}_3 \right) = \left(k_2 \boldsymbol{I}_3 + \hat{\boldsymbol{M}} \boldsymbol{Q} \right) \left(\hat{\boldsymbol{M}} \boldsymbol{Q} \boldsymbol{q}_1 - k_2 \boldsymbol{q}_2 + \boldsymbol{q}_3 \right) + \dot{\hat{\boldsymbol{M}}} \boldsymbol{Q} \boldsymbol{q}_2 + \boldsymbol{q}_2 + k_3 \boldsymbol{q}_3 \tag{26}$$

where $k_1,\ k_2,\ k_3 > 0$, and the update law for $\hat{\gamma}_1$ and $\hat{\gamma}_2$ is

$$\dot{\hat{\gamma}}_1 = k_\gamma \left(k_1 q_{21}^2 + (k_2 + \hat{\gamma}_1 k_1) k_1 q_{31} (q_{11} + q_{21}) \right) \tag{27a}$$

$$\dot{\hat{\gamma}}_2 = k_\gamma \left(k_1 q_{22}^2 + (k_2 + \hat{\gamma}_2 k_1) k_1 q_{32} (q_{12} + q_{22}) \right) \tag{27b}$$

where q_{ij} is the jth component of vector $\boldsymbol{q}_i$, and $k_\gamma > 0$. Then the error signals $\boldsymbol{e}_{yz}$, $\boldsymbol{q}_2$, $\boldsymbol{q}_3$ globally asymptotically converge to zero. This also implies the UAV flies along the horizontal parallel lines with desired velocity v_d.

Proof. The controller in (26) is obtained via a backstepping approach. We start the procedure with the image error kinematics (23). A Lyapunov function candidate L_1 is chosen as

$$L_1 = \frac{1}{2} \boldsymbol{e}_{yz}^T \boldsymbol{e}_{yz} \tag{28}$$

Then the time derivative of (28) is

$$\dot{L}_1 = -\frac{1}{Z^{v*}} \boldsymbol{e}_{yz}^T \boldsymbol{D}(\beta,\ f) \boldsymbol{L}_{yz}(\alpha^v) \boldsymbol{v}^v$$

Since Z^{v*} is positive, if the following equations holds

$$\boldsymbol{D}(\beta,\ f) \boldsymbol{L}_{yz}(\alpha^v) \boldsymbol{v}^v = k_1 \boldsymbol{e}_{yz}$$

it can be seen that system (23) is globally exponentially stable. In order to make $\boldsymbol{e}_{yz}$ converge to zero and the UAV fly at a desired velocity along the power lines, the velocity $\boldsymbol{v}^{v*}$ input should satisfy

$$\boldsymbol{H}(\beta,\ f) \boldsymbol{W}(\alpha^v) \boldsymbol{v}^{v*} = \boldsymbol{Q} \boldsymbol{q}_1$$

where $\boldsymbol{H}(\beta,\ f) = \mathrm{diag}([1,\ \beta f,\ 1])$. With the term $\boldsymbol{q}_2$ defined in (25b) describing the difference between the desired and the actual velocity, the time derivative of $\boldsymbol{q}_1$ and L_1 can be obtained as

$$\dot{\boldsymbol{q}}_1 = -\frac{1}{Z^{v*}} \begin{bmatrix} \boldsymbol{D}(\beta,\ f) \boldsymbol{L}_{yz}(\alpha^v) \\ \boldsymbol{0}_{1\times 3} \end{bmatrix} \boldsymbol{W}^{-1} \boldsymbol{H}^{-1}(\beta,\ \hat{f}) \boldsymbol{Q} (\boldsymbol{q}_2 + \boldsymbol{q}_1) = -\boldsymbol{M} \boldsymbol{Q} \left(\boldsymbol{q}_1 + \boldsymbol{q}_2 \right) \tag{29}$$

where $\boldsymbol{M} = \mathrm{diag}([\gamma_1,\ \gamma_2,\ 1])$, and

$$\begin{aligned} \dot{L}_1 &= -\frac{1}{Z^{v*}} \boldsymbol{e}_{yz}^T \boldsymbol{D}(\beta,\ f) \boldsymbol{L}_{yz}(\alpha^v) \boldsymbol{W}^{-1} \boldsymbol{H}^{-1}(\beta,\ \hat{f}) \boldsymbol{Q} (\boldsymbol{q}_2 + \boldsymbol{q}_1) \\ &= -\frac{1}{Z^{v*}} \boldsymbol{e}_{yz}^T \boldsymbol{D}(\beta,\ f) \boldsymbol{D}^{-1}(\beta,\ \hat{f}) k_1 \boldsymbol{e}_{yz} - \boldsymbol{q}_1^T \boldsymbol{M} \boldsymbol{Q} \boldsymbol{q}_2 \end{aligned} \tag{30}$$

Differentiating $\boldsymbol{W}$ in (25b) yields

$$\dot{\boldsymbol{W}} = \begin{bmatrix} 0 & 0 & 0 \\ \cos\alpha^v & -\sin\alpha^v & 0 \\ -\sin\alpha^v & -\cos\alpha^v & 0 \end{bmatrix} \dot{\alpha}^v = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha^v & \cos\alpha^v & 0 \\ \cos\alpha^v & -\sin\alpha^v & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\psi} = \boldsymbol{W} \boldsymbol{S}(\dot{\psi} \boldsymbol{E}_3) \tag{31}$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

Based on (25b), (29), (31), and the expression of $\boldsymbol{F}'$ in (25c), we can write

$$\begin{aligned}\dot{\boldsymbol{q}}_2 &= \boldsymbol{Q}^{-1}\boldsymbol{H}(\beta, \hat{f})\left(\dot{\boldsymbol{W}}\boldsymbol{v}^v + \boldsymbol{W}\dot{\boldsymbol{v}}^v\right) - \dot{\boldsymbol{q}}_1 \\ &= \boldsymbol{Q}^{-1}\boldsymbol{H}(\beta, \hat{f})\left(\boldsymbol{W}\boldsymbol{S}(\dot{\psi}\boldsymbol{E}_3)\boldsymbol{v}^v + \boldsymbol{W}\left(-\boldsymbol{S}(\dot{\psi}\boldsymbol{E}_3)\boldsymbol{v}^v + \frac{\boldsymbol{F}^v}{m}\right)\right) + \boldsymbol{M}\boldsymbol{Q}\left(\boldsymbol{q}_1 + \boldsymbol{q}_2\right) \\ &= \boldsymbol{F}' + \boldsymbol{M}\boldsymbol{Q}\left(\boldsymbol{q}_1 + \boldsymbol{q}_2\right)\end{aligned} \tag{32}$$

Continuing the backstepping method, let L_2 be the second Lyapunov function as

$$L_2 = L_1 + \frac{1}{2}\boldsymbol{q}_2^T\boldsymbol{q}_2 + \frac{1}{2k_\gamma}\left(\tilde{\gamma}_1^2 + \tilde{\gamma}_2^2\right) \tag{33}$$

where $\tilde{\gamma}_i = \gamma_i - \hat{\gamma}_i$, $i = 1, 2$. With (30) and (32), we obtain the time derivative of (33) as

$$\dot{L}_2 = -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta, f)\boldsymbol{D}^{-1}(\beta, \hat{f})k_1\boldsymbol{e}_{yz} + \boldsymbol{q}_2^T\left(\boldsymbol{F}' + \boldsymbol{M}\boldsymbol{Q}\boldsymbol{q}_2\right) - \frac{1}{k_\gamma}\left(\tilde{\gamma}_1\dot{\hat{\gamma}}_1 + \tilde{\gamma}_2\dot{\hat{\gamma}}_2\right) \tag{34}$$

If we choose

$$\boldsymbol{F}' = -k_2\boldsymbol{q}_2 - \hat{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 \tag{35}$$

then (34) can be written as

$$\dot{L}_2 = -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta, f)\boldsymbol{D}^{-1}(\beta, \hat{f})k_1\boldsymbol{e}_{yz} - k_2\boldsymbol{q}_2^T\boldsymbol{q}_2 + \boldsymbol{q}_2^T\tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 - \frac{1}{k_\gamma}\left(\tilde{\gamma}_1\dot{\hat{\gamma}}_1 + \tilde{\gamma}_2\dot{\hat{\gamma}}_2\right)$$

where $\tilde{\boldsymbol{M}} = \boldsymbol{M} - \hat{\boldsymbol{M}}$. Based on the definition of $\boldsymbol{q}_3$ in (25c) we solve for $\boldsymbol{F}'$ and substitute it into (32) to obtain

$$\dot{\boldsymbol{q}}_2 = \boldsymbol{M}\boldsymbol{Q}\boldsymbol{q}_1 - k_2\boldsymbol{q}_2 + \tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 + \boldsymbol{q}_3$$

and the time derivative of L_2 is

$$\dot{L}_2 = -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta, f)\boldsymbol{D}^{-1}(\beta, \hat{f})k_1\boldsymbol{e}_{yz} - k_2\boldsymbol{q}_2^T\boldsymbol{q}_2 + \boldsymbol{q}_2^T\tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 - \frac{1}{k_\gamma}\left(\tilde{\gamma}_1\dot{\hat{\gamma}}_1 + \tilde{\gamma}_2\dot{\hat{\gamma}}_2\right) + \boldsymbol{q}_2^T\boldsymbol{q}_3 \tag{36}$$

Using (35) the time derivative of $\boldsymbol{q}_3$ is

$$\begin{aligned}\dot{\boldsymbol{q}}_3 &= \dot{\boldsymbol{F}}' + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\left(\boldsymbol{M}\boldsymbol{Q}\boldsymbol{q}_1 - k_2\boldsymbol{q}_2 + \tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 + \boldsymbol{q}_3\right) + \dot{\hat{\boldsymbol{M}}}\boldsymbol{Q}\boldsymbol{q}_2 \\ &= \dot{\boldsymbol{F}}' + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\left(\hat{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_1 - k_2\boldsymbol{q}_2 + \boldsymbol{q}_3\right) + \dot{\hat{\boldsymbol{M}}}\boldsymbol{Q}\boldsymbol{q}_2 + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\tilde{\boldsymbol{M}}\boldsymbol{Q}\left(\boldsymbol{q}_2 + \boldsymbol{q}_1\right)\end{aligned} \tag{37}$$

Considering

$$L_3 = L_2 + \frac{1}{2}\boldsymbol{q}_3^T\boldsymbol{q}_3$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

and using (36) and (37)

$$\begin{aligned}\dot{L}_3 &= -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta,\, f)\boldsymbol{D}^{-1}(\beta,\, \hat{f})k_1\boldsymbol{e}_{yz} - k_2\boldsymbol{q}_2^T\boldsymbol{q}_2 + \boldsymbol{q}_2^T\tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 - \frac{1}{k_\gamma}\left(\tilde{\gamma}_1\dot{\hat{\gamma}}_1 + \tilde{\gamma}_2\dot{\hat{\gamma}}_2\right) + \boldsymbol{q}_2^T\boldsymbol{q}_3 \\ &\quad + \boldsymbol{q}_3^T\left(\dot{\boldsymbol{F}}' + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\left(\hat{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_1 - k_2\boldsymbol{q}_2 + \boldsymbol{q}_3\right) + \dot{\hat{\boldsymbol{M}}}\boldsymbol{Q}\boldsymbol{q}_2 + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\tilde{\boldsymbol{M}}\boldsymbol{Q}\left(\boldsymbol{q}_2 + \boldsymbol{q}_1\right)\right) \\ &= -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta,\, f)\boldsymbol{D}^{-1}(\beta,\, \hat{f})k_1\boldsymbol{e}_{yz} - k_2\boldsymbol{q}_2^T\boldsymbol{q}_2 \\ &\quad + \boldsymbol{q}_3^T\left(\dot{\boldsymbol{F}}' + \left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\left(\hat{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_1 - k_2\boldsymbol{q}_2 + \boldsymbol{q}_3\right) + \dot{\hat{\boldsymbol{M}}}\boldsymbol{Q}\boldsymbol{q}_2 + \boldsymbol{q}_2\right) \\ &\quad + \boldsymbol{q}_2^T\tilde{\boldsymbol{M}}\boldsymbol{Q}\boldsymbol{q}_2 - \frac{1}{k_\gamma}\left(\tilde{\gamma}_1\dot{\hat{\gamma}}_1 + \tilde{\gamma}_2\dot{\hat{\gamma}}_2\right) + \boldsymbol{q}_3^T\left(k_2\boldsymbol{I}_3 + \hat{\boldsymbol{M}}\boldsymbol{Q}\right)\tilde{\boldsymbol{M}}\boldsymbol{Q}\left(\boldsymbol{q}_2 + \boldsymbol{q}_1\right)\end{aligned}$$

From the definition of $\boldsymbol{R}_{\theta\phi}$, we can get its the time derivative as

$$\dot{\boldsymbol{R}}_{\theta\phi} = \boldsymbol{R}_{\theta\phi}\boldsymbol{S}\left(\boldsymbol{\omega}^c\right) - \boldsymbol{S}\left(\dot{\psi}\boldsymbol{E}_3\right)\boldsymbol{R}_{\theta\phi} \tag{38}$$

Then from the expression of $\boldsymbol{F}'$, (31), and (38) we can write

$$\dot{\boldsymbol{F}}' = -\boldsymbol{\Delta}_1\boldsymbol{W}\boldsymbol{R}_{\theta\phi}\left(\boldsymbol{S}(\boldsymbol{\omega}^c)T\boldsymbol{E}_3 + \dot{T}\boldsymbol{E}_3\right)$$

It can be seen that if we choose the control law (26), and the parameter update law (27), the time derivative of L_3 can be written as

$$\dot{L}_3 = -\frac{1}{Z^{v*}}\boldsymbol{e}_{yz}^T\boldsymbol{D}(\beta,\, f)\boldsymbol{D}^{-1}(\beta,\, \hat{f})k_1\boldsymbol{e}_{yz} - k_2\boldsymbol{q}_2^T\boldsymbol{q}_2 - k_3\boldsymbol{q}_3^T\boldsymbol{q}_3$$

Hence, we can conclude that the system is globally asymptotically stable. □

In the above proof an additional backstepping iteration could be performed to obtain a control law for torque. However, such an approach would unnecessarily complicate the design. The IMU provides a relatively high frequency measurement of $\boldsymbol{\omega}^c$ compared to the low frequency video measurement. Hence, this time scale separation allows for a high bandwidth inner loop for $\boldsymbol{\omega}^c$. The inner loop performance can be treated as ideal from the perspective of the outer loop IBVS design. A similar two-loop structure was proposed in[25] where experimental validation was performed.

We remark that (26) appears to define the four inputs, i.e., three angular velocity inputs and thrust. However, expanding (26) reveals only expressions for the first and second components of $\boldsymbol{\omega}^c$ are determined. The third component is given by the yaw control law in the next subsection.

5.2. Yaw rate control

In order to make e_α converge to zero we design the yaw rate of the UAV as

$$\dot{\psi} = -k_\alpha e_\alpha = -k_\alpha s_3 \tag{39}$$

with $k_\alpha > 0$. Since the Euler rates and $\boldsymbol{\omega}^c$ are related by

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & s_\phi s_\theta/c_\theta & c_\phi s_\theta/c_\theta\\ 0 & c_\phi & -s_\phi\\ 0 & s_\phi/c_\theta & c_\phi/c_\theta\end{bmatrix}\boldsymbol{\omega}^c$$

Hence, the third component of the angular velocity fed to the inner loop is given as

$$\omega_3^c = \frac{c_\theta}{c_\phi}\left(-k_\alpha s_3 - \omega_2^c\frac{s_\phi}{c_\theta}\right) \tag{40}$$

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

Table I. Physical parameters.

$\boldsymbol{J}$	$\text{diag}\{0.01,\ 0.01,\ 0.05\}\ \text{kg}\cdot\text{m}^2$
m	1 kg
f	500 pixels
Pixel size	$10\,\mu$m
Image size	[1280, 1024]
Image centre $[u_0, v_0]$	[640, 512]

Table II. Control gains.

k_1	10
k_2	5
k_3	2
k_α	0.6
k_γ	1
β	$\frac{1}{50}$

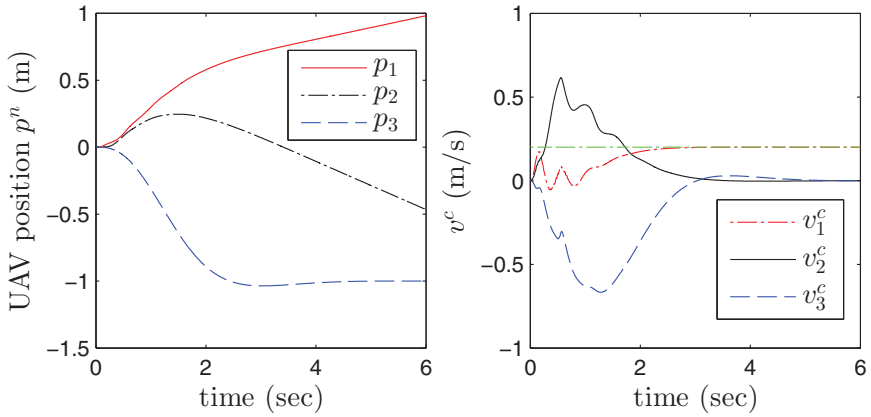


Fig. 4. Time evolution of UAV's position $\boldsymbol{p}^n$ and velocity $\boldsymbol{v}^c$.

The dynamic IBVS feedback law T and $\boldsymbol{\omega}^c$ is given by (26) and (40). As mentioned above, (26) defines the control for the first two components of $\boldsymbol{\omega}^c$ and (40) provides the law for ω_3^c.

6. Simulation

In this section the proposed IBVS control law with line feature is tested in simulation. The dynamic IBVS control law update frequency is 30 Hz. The inner loop PID control for attitude is updated at the frequency of 200 Hz which is typical of an IMU. The reference for the inner loop is produced by the IBVS controller. The physical parameters used in the simulation are shown in Table I. They are representative of a quadrotor test stand developed at the Applied Nonlinear Control Lab at the University of Alberta.

We define $N = 4$ lines in the navigation frame $\mathcal{N}$ as

$$\begin{aligned} L_1 &: \ 2X + Y = 1,\ Z = 4 \\ L_2 &: \ 2X + Y = -1,\ Z = 4 \\ L_3 &: \ 2X + Y = 2,\ Z = 4 \\ L_4 &: \ 2X + Y = 0,\ Z = 4 \end{aligned}$$

The control gains and scaling coefficient β are given in Table II.

We denote $\boldsymbol{p}^n = [p_1\ p_2\ p_3]^T$ as the UAV's position expressed in navigation frame $\mathcal{N}$ and take the initial states of the vehicle $\boldsymbol{p}^n(0)$, $\boldsymbol{v}^v(0)$, $\boldsymbol{\omega}^b(0)$, $\boldsymbol{\omega}^b(0)$ as zero. Also, we take the initial parameter

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

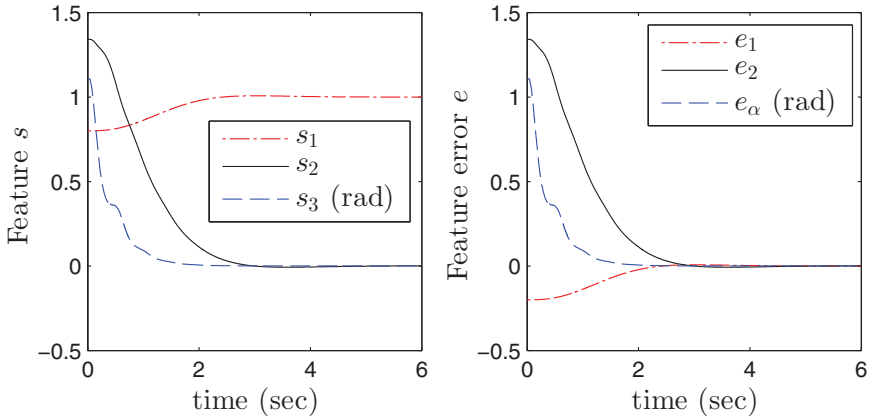


Fig. 5. Time evolution of Line Feature $\boldsymbol{s}$ and its error $\boldsymbol{e}$.

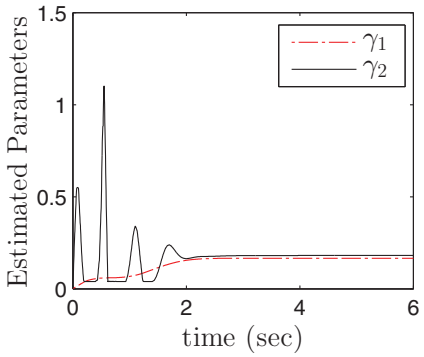


Fig. 6. Time evolution of $\hat{\gamma}_i, i = 1, 2$.

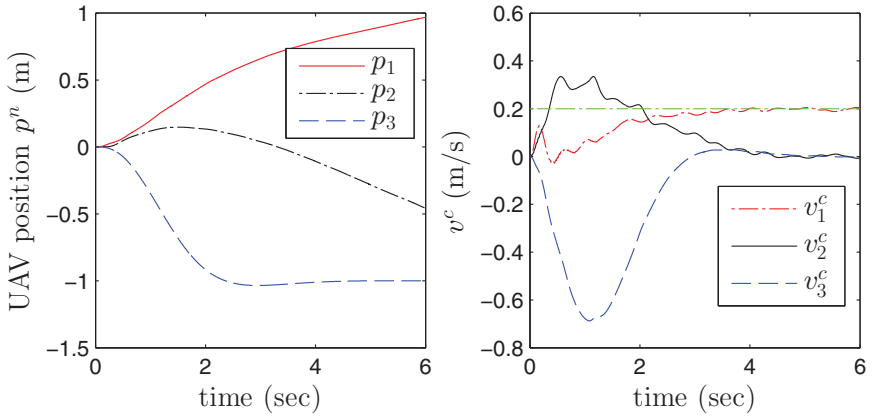


Fig. 7. Time evolution of UAV's position $\boldsymbol{p}^n$ and velocity $\boldsymbol{v}^c$ with attitude measurement noise $\hat{f} = 1.5f$.

estimates $\hat{\gamma}_i = 0, i = 1, 2$, and $\boldsymbol{R}_C^N(0) = \boldsymbol{I}$. The reference roll ϕ and pitch θ are bounded by $\pm 10^\circ$. The desired velocity along the parallel lines is set as $v_d = 0.2$ m/s and the μ_{20}^{v*} is chosen to correspond to a UAV position with $p_3 = -1$. That is, the UAV should move down by 1 m to achieve the desired s_1^*.

First, we assume focal length is perfectly calibrated. Figure 4 shows the resulting trajectory of the UAV in the navigation frame $\mathcal{N}$, and the translational velocity expressed the camera frame $\mathcal{C}$. The trajectories of the proposed line feature and its image error are shown in Fig. 5. It can been seen that the altitude of the UAV decreases to its desired value of -1 m in a few seconds. This makes s_1 converges to s_1^*. Also, the desired velocity of $v_d = 0.2$ m/s is achieved. As expected, Fig. 5 shows

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

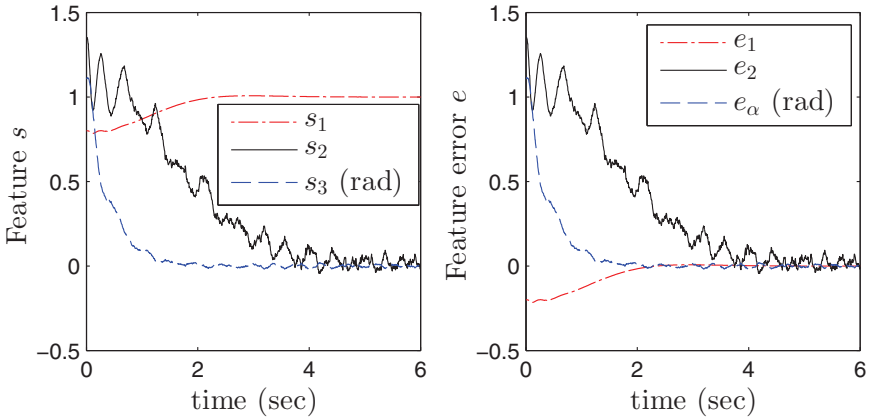


Fig. 8. Time evolution of Line Feature $\boldsymbol{s}$ and its error $\boldsymbol{e}$ with attitude measurement noise and $\hat{f} = 1.5f$.

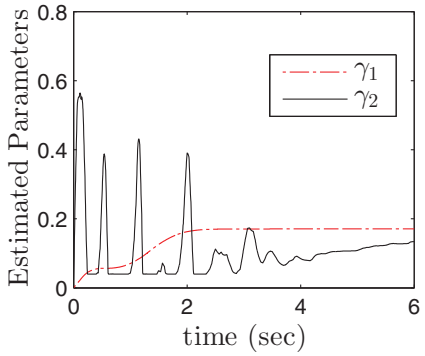


Fig. 9. Time evolution of $\hat{\gamma}_i, i = 1, 2$ with attitude measurement noise and $\hat{f} = 1.5f$.

that the image errors converge to zero. Figure 6 shows the estimated value $\hat{\gamma}_i, i = 1, 2$ converges to a constant value.

In order to check the proposed approach's robustness to calibration error of intrinsic camera parameters and measurement noise in attitude, we set the estimated focal length as $\hat{f} = 1.5f$ and add noise to the attitude with a spectral power of 2.5×10^{-6}, i.e., roughly a magnitude of 2°. Before using the measured attitude to convert the real image feature to the virtual image feature, a second order low pass filter is used to filter the attitude signal. The simulation results with noise corrupted attitude measurement are shown in Figs. 7, 8, and 9. In spite of the presence of noise and calibration error the performance of the proposed IBVS controller is satisfactory with trajectories for $\boldsymbol{p}^n$ and $\boldsymbol{v}^c$ being similar to the ideal case in Fig. 4.

Assumption 1 restricts the 3D target line to a horizontal plane. In some scenarios this assumption may not hold, and we simulate the case where the lines lie in a plane rotated 20° about $\boldsymbol{n}_1$. The same noise level and error in estimated focal length was considered. The results are shown in Figs. 10 and 11. As shown in Fig. 10, unlike previous cases the UAV keeps moving upwards because it controls its height above the inclined plane to a constant to achieve the desired value s_1^*. We can also see from Fig. 11 that the image feature errors converge to zero even though the 3D target lines lie in a non-horizontal plane.

7. Conclusion

A new IBVS control design based on image features from projected lines in a virtual camera image has been presented in this paper. The method for converting projected lines in the real camera image to lines in the virtual camera image has been provided. Under the assumption that all 3D parallel lines are horizontal, the image error kinematics has been shown to have a simple decoupled

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

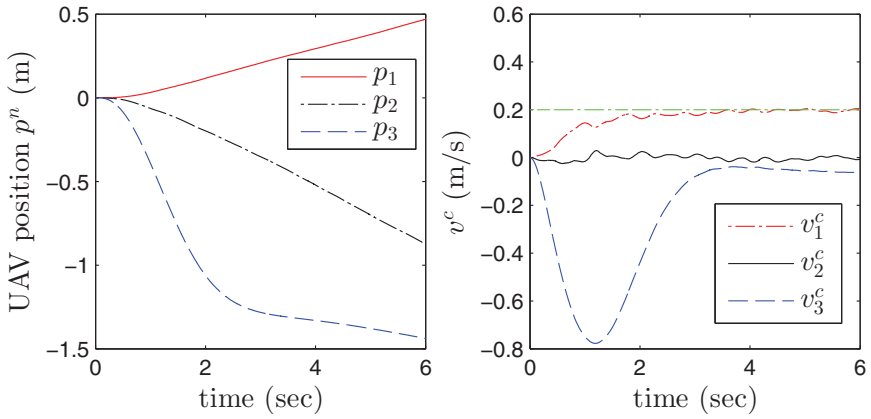


Fig. 10. Time evolution of UAV's position $\boldsymbol{p}^n$ and velocity $\boldsymbol{v}^c$ with attitude measurement noise, $\hat{f} = 1.5f$ and non-horizontal target plane.

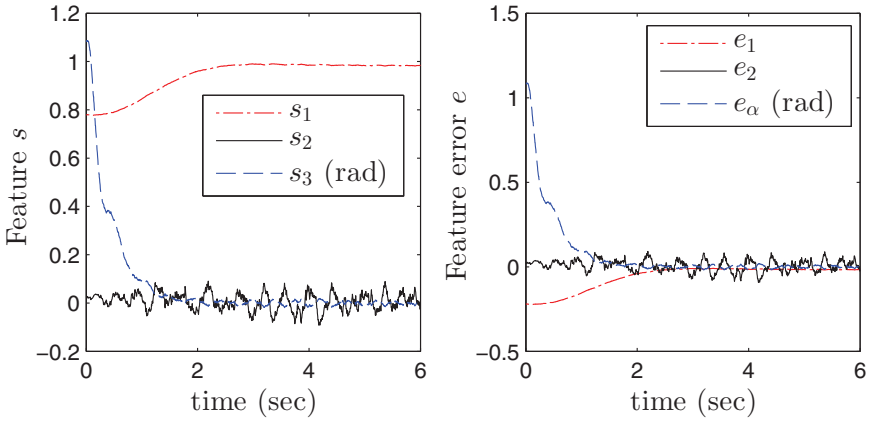


Fig. 11. Time evolution of Line Feature $\boldsymbol{s}$ and its error $\boldsymbol{e}$ with attitude measurement noise, $\hat{f} = 1.5f$ and non-horizontal target plane.

structure. The proof of global asymptotic stability of the error dynamics is given. The proposed IBVS control does not require any depth estimation and is robust to camera calibration errors. Although the simulation results presented here demonstrate that the design method still works with lines in a non-horizontal plane, no rigourous proof of this robustness is provided. Future work will focus on removing Assumption 1 so that the parallel lines are constrained to planes with more general orientation (i.e., non-horizontal). Experimental work is also planned to validate the IBVS control scheme in practice.

References

1. F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *J. Field Robot.* **29**(2), 315–378 (2012).
2. R&D Program Office, System Planning and Asset Management Div., BCTC, *Transmission Technology Roadmap: Pathways to BC's Future Grid* (2008).
3. C. Whitworth, A. Duller, D. Jones and G. Earp, "Aerial video inspection of overhead power lines," *Power Eng. J.* **15**, 25–32 (Feb. 2001).
4. J. R. Azinheira and P. Rives, "Image-based visual servoing for vanishing features and ground lines tracking: Application to a UAV automatic landing," *Int. J. Optomechatronics* **2**(3), 275–295 (2008).
5. F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robot. Autom. Mag.* **13**(4), 82–90 (2006).
6. F. Chaumette and S. Hutchinson, "Visual servo control part II: Advanced approaches," *IEEE Robot. Autom. Mag.* **14**(1), 109–118 (2007).

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press

7. E. Rondon, L.-R. Garcia-Carrillo and I. Fantoni, "Vision-Based Altitude, Position and Speed Regulation of a Quadrotor Rotorcraft," *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Taipei, Taiwan) (Oct. 2010) pp. 628–633.
8. L. Carrillo, G. Flores Colunga, G. Sanahuja and R. Lozano, "Quad rotorcraft switching control: An application for the task of path following," *IEEE Trans. Control Syst. Technol.* **22**(4), 1255–1267 (2014).
9. B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," **In**: *Experimental Robotics III* (T. Yoshikawa and F. Miyazaki, eds.) *Lecture Notes in Control and Information Sciences*, vol. 200 (Springer, Berlin, 1994) pp. 182–192.
10. P. Corke and M. Good, "Dynamic Effects in High-Performance Visual Servoing," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France (May 1992) pp. 1838–1843.
11. T. Komuro, A. Iwashita and M. Ishikawa, "A QVGA-size pixel-parallel image processor for 1,000-FPS vision," *IEEE Micro* **29**(6), 58–67 (2009).
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), 187–198 (2002).
13. 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.* **25**(3), 743–749 (2009).
14. R. Mahony and T. Hamel, "Image-based visual servo control of aerial robotic systems using linear image features," *IEEE Trans. Robot.* **21**(2), 227–239 (2005).
15. N. Metni, T. Hamel and F. Derkx, "Visual Tracking Control of Aerial Robotic Systems with Adaptive Depth Estimation," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain (Dec. 2005) pp. 6078–6084.
16. H. de Plinval, P. Morin, P. Mouyon and T. Hamel, "Visual Servoing for Underactuated VTOL UAVs: A Linear, Homography-Based Approach," *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China (May 2011) pp. 3004–3010.
17. 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* **24**(16), 2285–2308 (2014).
18. R. Ozawa and F. Chaumette, "Dynamic Visual Servoing with Image Moments for a Quadrotor using a Virtual Spring Approach," *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China (May 2011) pp. 5670–5676.
19. 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), 683–696 (2013).
20. H. Jabbari, G. Oriolo and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated UAV," *Int. J. Robot. Autom.* **29**(1), 92–104 (2014).
21. B. Espiau, F. Chaumette and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.* **8**(3), 313–326 (1992).
22. B. Godbolt, Experimental Nonlinear Control of a Helicopter Unmanned Aerial Vehicle (UAV) *Ph.D. Thesis* (Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2013).
23. A. De Luca, G. Oriolo and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. J. Robot. Res.* **27**(10), 1093–1116 (2008).
24. O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. Robot.* **21**(6), 1116–1127 (2005).
25. N. Guenard, T. Hamel and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.* **24**(2), 331–340 (2008).

https://doi.org/10.1017/S0263574714002707 Published online by Cambridge University Press