

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/277722689>

Monocular distance estimation with optical flow maneuvers and efference copies: A stability-based strategy

Article in *Bioinspiration & Biomimetics* · January 2016

DOI: 10.1088/1748-3190/11/1/016004 · Source: arXiv

CITATIONS

19

READS

187

1 author:



[Guido de Croon](#)

Delft University of Technology

145 PUBLICATIONS 1,211 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



DelFly [View project](#)



High speed aerial navigation using event-based cameras [View project](#)

Bioinspiration & Biomimetics



PAPER

OPEN ACCESS

RECEIVED
3 June 2015

REVISED
3 November 2015

ACCEPTED FOR PUBLICATION
20 November 2015

PUBLISHED
7 January 2016

Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy

Guido C H E de Croon

Micro Air Vehicle Laboratory, Control and Simulation, Faculty of Aerospace Engineering, TU Delft, the Netherlands

E-mail: g.c.h.e.decroon@tudelft.nl

Keywords: optical flow, distance estimation, efference copies, micro air vehicle

Abstract

The visual cue of optical flow plays an important role in the navigation of flying insects, and is increasingly studied for use by small flying robots as well. A major problem is that successful optical flow control seems to require distance estimates, while optical flow is known to provide only the ratio of velocity to distance. In this article, a novel, stability-based strategy is proposed for monocular distance estimation, relying on optical flow maneuvers and knowledge of the control inputs (efference copies). It is shown analytically that given a fixed control gain, the stability of a constant divergence control loop only depends on the distance to the approached surface. At close distances, the control loop starts to exhibit self-induced oscillations. The robot can detect these oscillations and hence be aware of the distance to the surface. The proposed stability-based strategy for estimating distances has two main attractive characteristics. First, self-induced oscillations can be detected robustly by the robot and are hardly influenced by wind. Second, the distance can be estimated during a zero divergence maneuver, i.e., around hover. The stability-based strategy is implemented and tested both in simulation and on board a Parrot AR drone 2.0. It is shown that the strategy can be used to: (1) trigger a final approach response during a constant divergence landing with fixed gain, (2) estimate the distance in hover, and (3) estimate distances during an entire landing if the robot uses adaptive gain control to continuously stay on the ‘edge of oscillation.’

1. Introduction

A major challenge in robotics is to achieve autonomous operation of tiny flying robots such as 25 g ‘pocket drones’ [11] or more extremely the 80 mg Robobee [33]. Flying insects provide a rich source of inspiration for solving this challenge, since they are able to navigate successfully on the basis of a very limited sensory and processing apparatus. Flying insects rely heavily on optical flow, i.e. the apparent motion of world points caused by the relative motion between the observer and the environment [21]. Also for robots this cue is promising as it can be extracted from a single passive camera, implying light weight and relatively little energy consumption [14, 16, 18, 41].

It is well known that optical flow conveys information on the ratio between distance and velocity, and that additional information is necessary to disentangle these quantities. This ‘scaling’ can be performed by

means of maneuvers that have a known range or velocity (as in parallax [29]), or with any sensor that directly or indirectly provides information on the (change in) distance or velocity estimates, e.g. accelerometers or sonar [12, 32]. Interestingly, flying insects such as fruit flies do not seem to have scaling sensors [15]. In addition, it is unlikely that they know the range of their maneuvers since they only have access to their air speed, not their ground speed [6].

Still, flying insects are able to navigate successfully. The key to this success is that flying insects follow straightforward strategies based on optical flow observables. First, it was found that honeybees keep ventral flow (defined as $\vartheta_x = \frac{v_x}{z}$) constant when they perform grazing landings [42]. This strategy was studied for flying robots and simulated spacecraft [27, 35, 37, 45]. However, these studies required a ‘pitching’ law, ultimately depending on an estimate of the height and velocity. The main reason seemed to be the absence of

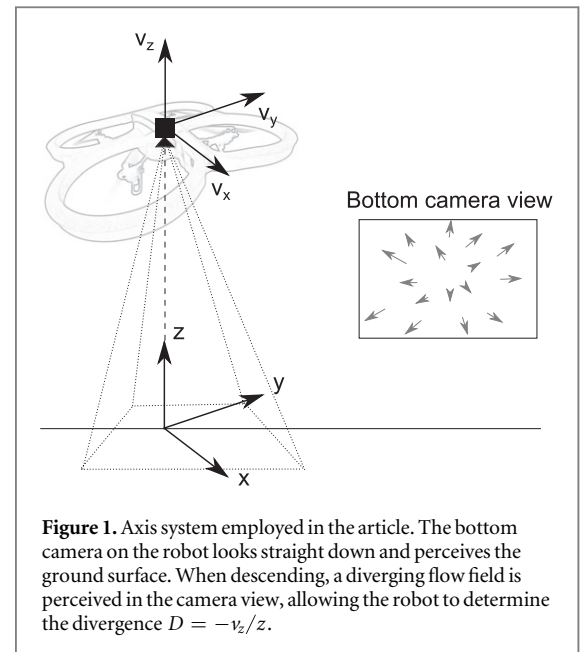
control of the vertical dynamics. Therefore, more recent landing strategies also include the flow divergence (D , typically defined as $\frac{-v_z}{z}$, with the positive z , v_z axis pointing up) or time-to-contact ($\tau = \frac{-z}{v_z}$) [1, 24, 26, 28]. These strategies either enforce a decreasing time-to-contact (as humans do for car braking [30]) or keep the divergence constant (as honeybees do for straight landings [3]). Recently, in [15], ventral flow and divergence were combined with slope estimation and an anemometer in order to fly over uneven terrain, without a need for inertial frames or scaling sensors.

The absence of velocity and distance information has been a defining property of the work on optical flow-based navigation strategies. However, the following observations suggest that for successful navigation distance estimates are still of importance. First, flying insects do seem to have distance-dependent behaviors. For instance, van Breugel *et al* [6] remarked that fruit flies extend their legs at a specific distance from an oncoming object. Second, for successful optical flow landing control, the gains of the controller are tuned to a range of specific heights and velocities. For instance, for the constantly decreasing time-to-contact landings in [9, 28], the gains are scheduled by means of τ , along a trajectory starting at a specific initial height and velocity. Third, while for a constantly decreasing time-to-contact landing it is possible to perform a final landing procedure (when the time-to-contact is almost zero), it is not obvious how such a procedure should be triggered in a constant time-to-contact or divergence landing.

It is actually possible to estimate distances based on optical flow, if a robot has access to copies of its control input signals (referred to as ‘efference copies’ in the biological literature). Two main strategies have been proposed so far in the literature.

The first strategy for estimating distances is well-established in the field of image based visual servoing (IBVS). In the task of IBVS a robot has to regulate certain target features to desired image locations, without knowing *a priori* their dimensions or geometry. Quite early it was observed that if a fixed gain is used to map an image error to a control input, the response of the system depends on the distance to the target [8]. Given a model of the robot’s dynamics, the effects of the robot’s actions on the image features can be used to estimate the distance to the target (e.g. with a non-linear observer). The distance estimates allow for smooth control in an adaptive gain control scheme [2].

The second strategy for estimating distances was recently proposed by [6]. They showed that during a constant divergence approach of an object, the control input u (the thrust) is proportional to the distance d , and that efference copies can then be used as a stand-in for distance. This solution was tested on a camera mounted on a rail and successful distance estimates



were obtained by integrating information during the entire approach.

Both strategies above assume that the robot’s accelerations are completely determined by the control inputs. However, flying robots and insects are also subject to significant external accelerations caused by wind or wind gusts.

The **main contribution** of this article is the proposition of a novel, stability-based, strategy for distance estimation with optical flow and efference copies. The central idea is to estimate distance by exploiting the self-induced oscillations that result from the fundamental imperfection of fixed-gain optical flow-based control. The distance estimation strategy provides two major advantages with respect to the previously proposed strategies: (1) it works straightforwardly even in the presence of wind, and (2) it can be used in a broader range of maneuvers, and notably in hover.

The remainder of the article is organized as follows. In section 2, we introduce the notations used in this article and discuss the intrinsics of a constant divergence landing. Subsequently, the new stability-based strategy to distance estimation is proposed in section 3. In section 4 it is shown how self-induced oscillations can be used to trigger a final landing procedure. Subsequently, in section 5 adaptive gain control is introduced to determine the distance in hover and to estimate distances along an entire constant divergence landing. In section 6, the findings are empirically verified with real-world experiments using a Parrot AR drone. Finally, the results are discussed in section 7 and conclusions are drawn in section 8.

2. Constant divergence landing

The novel distance estimation strategy will be studied in the context of a landing task. In this section, the

main aspects of a constant divergence landing are explained. Figure 1 shows the definition of the axes as used in the formulas. In order to keep the equations as uncluttered as possible, most of the analysis in this article focuses purely on the z -axis. Generalizations to other looking and movement directions are easily made (see, e.g., appendix B), as are the inclusion of different attitudes or a displacement and rotation of the camera with respect to the body center of mass.

The robot perceives an optical flow field \vec{q} in its bottom camera view. The divergence is defined here as: $D = \frac{1}{2} \nabla \vec{q}$, which is equal to $-\frac{v_z}{z}$ if the robot is looking straight down. In this article as ‘visual observable’ the variable ϑ_z is used, which is the relative velocity $\vartheta_z = \frac{v_z}{z} = -D$. Hence, a constant or zero divergence maneuver is also a constant or zero ϑ_z maneuver.

If the robot lands while keeping ϑ_z constant, the height and vertical velocity will obey the formulas:

$$z(t) = z_0 e^{(\vartheta_{z0}/z_0)t}, \quad (1)$$

$$v_z(t) = v_{z0} e^{(\vartheta_{z0}/z_0)t}, \quad (2)$$

which with a negative initial velocity v_{z0} has v_z and z simultaneously converge to 0, $\frac{v_z(t)}{z(t)} = \frac{v_{z0}}{z_0} = -c^2$.

3. Instability of constant divergence landings

The onset of the work in this article came from the difficulty of making optical flow-based landing strategies work on real micro air vehicles (MAVs). The study in [28] is especially instructive, since multiple control laws are studied, of which the gains depend to different extents on the initial height and velocity. Especially important are the remarks and observations on the difficulty of getting the optical flow control to work close to the landing surface, where oscillations tend to occur. It is even suggested to disengage the optical flow control under a threshold distance (as is also done in [45]).

Please note that it is not easy to identify the cause of oscillations close to the surface when performing vision-based control with a real MAV. Oscillations close to the ground can potentially be explained by the ‘ground effect’ due to the MAV’s downwash and by the vision process (more blurry images, larger optical flow vectors that are harder to track). Below it will become clear though that even without such additional effects, the instability problem arises.

Based on the dependence of the gains on the actual height and velocity, in this article a novel strategy is proposed for distance estimation: *stability-based* distance estimation. In this section, first the fundamental reason for the gain tuning problem of optical flow based control is explained (section 3.1). Subsequently,

a specific control law is assumed and it is shown analytically that the stability directly relates the control gain to the height (section 3.2).

3.1. Fundamental reason for gain tuning

Before going into why a single gain is only stable for a given range of heights, first the equations of motion will be described. Let us start with the model in [6]. In the notation of a state space model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (3)$$

where the state vector $\mathbf{x} = [z(t), v_z(t)]^T$, and $\mathbf{u}(t) = u_z(t)$, so that:

$$\begin{bmatrix} \dot{z}(t) \\ \dot{v}_z(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z(t) \\ v_z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_z(t). \quad (4)$$

This model assumes that $u_z = a_z$, which effectively implies that any possible gravity is subsumed under the u_z -term:

$$a_z = -g + \frac{u'_z}{m} = u_z, \quad (5)$$

where $u'_z = m(a_z + g)$ is the actual upward thrust generated by the robot in Newton. It is common in control system theory to subsume factors such as gravity and the mass into u_z as the calculation of u'_z is straightforward and does not depend on any state variables.

Equation (5) is only valid if the control force is the only force acting on the robot. In a vacuum environment this can be approximately correct, e.g. for a moon landing. However, a robot flying in the air will undergo additional accelerations. Therefore, in this article, the drag force will be modeled, which depends on the robot’s movement relative to the air:

$$f_D = \text{sign}(v_{\text{air}}) \frac{1}{2} \rho C_D A v_{\text{air}}^2, \quad (6)$$

where:

$$v_{\text{air}} = v_{\text{wind}} - v_z, \quad (7)$$

and $\text{sign}(v_{\text{air}})$ indicates the directionality of f_D along the z -axis. This leads to an additional acceleration:

$$a_z = -g + \frac{u'_z + f_D}{m}, \quad (8)$$

in which f_D is a time-varying value involving a non-linear function of v_z and an uncontrolled variable v_{wind} .

Now, let us start from the visual observable ϑ_z :

$$\vartheta_z = \frac{v_z}{z}. \quad (9)$$

If ϑ_z is differentiated over time, it results in:

$$\dot{\vartheta}_z = \frac{a_z}{z} - \frac{v_z^2}{z^2}. \quad (10)$$

Using equation (9):

$$\dot{\vartheta}_z = \frac{a_z}{z} - \vartheta_z^2. \quad (11)$$

Assuming a_z as in equation (8):

$$\dot{\vartheta}_z = \frac{-g + \frac{u'_z + f_D}{m}}{z} - \vartheta_z^2. \quad (12)$$

Differentiating this formula with respect to u'_z gives:

$$\frac{\partial \dot{\vartheta}_z}{\partial u'_z} = \frac{1}{mz}. \quad (13)$$

Equation (13) shows that a change in thrust has a larger effect on the change in $\dot{\vartheta}_z$ closer to the landing surface than far away (independently of external accelerations)¹ Thus, a control gain for regulating ϑ_z that leads to a satisfactory control performance far away from the landing surface, will lead to very large effects on $\dot{\vartheta}_z$ and hence indirectly on ϑ_z close to the landing surface. Although theoretically one could use $\frac{\partial \dot{\vartheta}_z}{\partial u'_z}$ directly for distance estimation ($z = \frac{\partial u'_z}{\partial \vartheta_z} \frac{1}{m}$), in practice this value is extremely noisy, since it involves a partial derivative with the noisy value $\dot{\vartheta}_z$.

3.2. Relation between control stability, gain, and height

In this subsection, it will be shown that the larger influence of u_z on ϑ_z at lower heights eventually leads to instability at a specific height. For the analysis, the following constant divergence control law is studied, using just a proportional gain K_z as in [6]:

$$u_z = K_z(\vartheta_z^* - \vartheta_z), \quad (14)$$

where it is easy to see that in a noiseless, delayless system:

$$\lim_{K_z \rightarrow \infty} \frac{u_z}{K_z} = (\vartheta_z^* - \vartheta_z) = 0, \quad (15)$$

and hence $\vartheta_z = \vartheta_z^*$. Real systems always have some noise and delay. Below, it will be shown that just discretizing the control with zero-order-hold (ZOH) will already introduce an instability into the system.

Let us assume the state space model of equation (4), ignoring drag for the moment. The corresponding observation is:

$$y = \vartheta_z = \frac{v_z}{z}, \quad (16)$$

which is a nonlinear function. Linearizing the state space model gives:

$$\begin{aligned} \Delta y(t) &= C \Delta x(t) + D \Delta u(t) \\ &= \left[\frac{-v_z}{z^2} \quad \frac{1}{z} \right] \begin{bmatrix} \Delta z(t) & \Delta v_z(t) \end{bmatrix}^\top, \end{aligned} \quad (17)$$

¹ The constancy of this relation depends on the constancy of the robot's mass, which applies for landing quad rotors but not for spacecraft. A spacecraft will loose more and more mass when landing, hence aggravating the effect in equation (13).

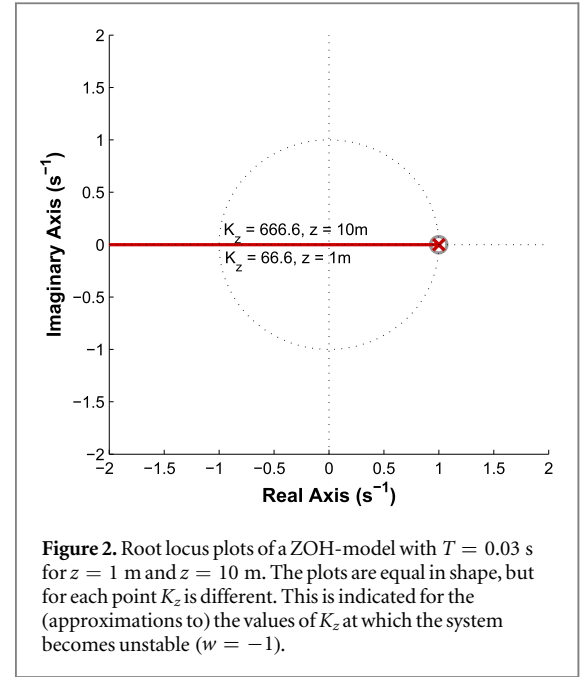


Figure 2. Root locus plots of a ZOH-model with $T = 0.03$ s for $z = 1$ m and $z = 10$ m. The plots are equal in shape, but for each point K_z is different. This is indicated for the (approximations to) the values of K_z at which the system becomes unstable ($w = -1$).

so that the state space model matrices are:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} -v_z & 1 \\ z^2 & z \end{bmatrix} \quad D = [0]. \quad (18)$$

As mentioned, the continuous system without noise and delay is stable (equation (15)). Here, the discretized system is studied, which has the following state space model matrices corresponding to the continuous ones in equation (18):

$$\begin{aligned} \Phi &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \\ C &= \begin{bmatrix} -v_z & 1 \\ z^2 & z \end{bmatrix} \quad D = [0], \end{aligned} \quad (19)$$

where T is the discrete time step in seconds. The transfer function of the open loop system can be determined to be:

$$G(w) = C(wI - \Phi)^{-1}\Gamma, \quad (20)$$

$$= \frac{\left(zT - \frac{1}{2}v_z T^2 \right) w - zT - \frac{1}{2}v_z T^2}{z^2(w - 1)^2}, \quad (21)$$

where we use w as the Z-transform variable, since z already represents height. The feedback transfer function is:

$$G(w) = \frac{K_z \left(\left(zT - \frac{1}{2}v_z T^2 \right) w - zT - \frac{1}{2}v_z T^2 \right)}{z^2(w - 1)^2 + K_z \left(\left(zT - \frac{1}{2}v_z T^2 \right) w - zT - \frac{1}{2}v_z T^2 \right)}. \quad (22)$$

Equation (22) shows that, given a gain K_z and time step T , the dynamics and stability of the system depend both on the height z and velocity v_z .

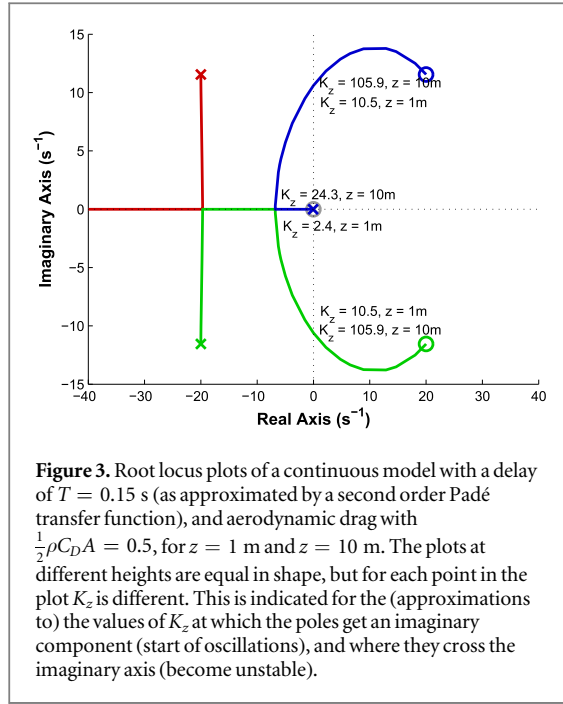


Figure 2 is the root locus plot of $G(w)$. For the Z -transform any pole in the unit circle is a stable pole. The different elements of the plot can be found back in equation (22). The numerator indicates a single finite zero at:

$$w_0 = \frac{zT + \frac{1}{2}v_z T^2}{zT - \frac{1}{2}v_z T^2}. \quad (23)$$

Given $z > 0$, $T > 0$, and $v_z < 0$, w_0 is positive and due to the typically small T values slightly smaller than 1. There is also a negative infinite zero. Since the denominator of equation (22) is a second order equation in w , $G(w)$ has two poles, both of which are stable. For $K_z = 0$, the poles are located at $w = 1$. As the gain K_z increases, one pole moves toward the finite zero and one toward the infinitely negative zero. This latter pole is interesting, because as soon as it passes $w = -1$, the system becomes unstable. Setting $w = -1$ in the denominator and equating it with 0 gives an equation for the unstable gain value K_z :

$$K_z = \frac{2}{T}z. \quad (24)$$

This implies that given a specific gain K_z and time step T , there is always a height at which the control system will become unstable:

$$z = \frac{1}{2}K_z T. \quad (25)$$

When using a fixed gain, a tradeoff has to be made between the control performance at a larger height (requiring a large gain) and at lower heights (requiring a small gain). Please remark that the instability at lower altitudes is exactly what is observed for robotic systems performing constant divergence landings!

3.3. Including wind

In this subsection it is shown that also for a model with wind, the instability of the system given a K_z depends on z . Including wind leads to a changed formula for \dot{v}_z (see equations (6) and (8)):

$$\dot{v}_z = u_z + \text{sign}(v_{\text{wind}} - v_z) \frac{1}{2m} \rho C_D A (v_{\text{wind}} - v_z)^2, \quad (26)$$

where from now on β will stand in for the constant $\frac{1}{2m} \rho C_D A$. In order to obtain a linear state space model, this equation is linearized to obtain:

$$\Delta \dot{v}_z = \Delta u_z - \left(\text{sign}(v_{\text{wind}} - v_z) \times \beta (v_{\text{wind}} - v_z) \right) \Delta v_z, \quad (27)$$

where the factor multiplied with Δv_z is a constant in the linearized model (v_{wind} and v_z are given the values at the linearization point). In order to avoid cluttering the formulas, this constant is represented by $p = \text{sign}(v_{\text{wind}} - v_z) \beta (v_{\text{wind}} - v_z)$ (where $p > 0$). This leads to the following continuous, linear state space model:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -p \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} -v_z & 1 \\ z^2 & z \end{bmatrix} \quad D = [0], \quad (28)$$

which results in the Z -transform matrices:

$$\Phi = \begin{bmatrix} 1 & \frac{(1 - e^{-pT})}{p} \\ 0 & e^{-pT} \end{bmatrix} \quad \Gamma = \begin{bmatrix} \frac{T}{p} - \frac{(1 - e^{-pT})}{p^2} \\ \frac{(1 - e^{-pT})}{p} \end{bmatrix} \quad (29)$$

$$C = \begin{bmatrix} -v_z & 1 \\ z^2 & z \end{bmatrix} \quad D = [0], \quad (30)$$

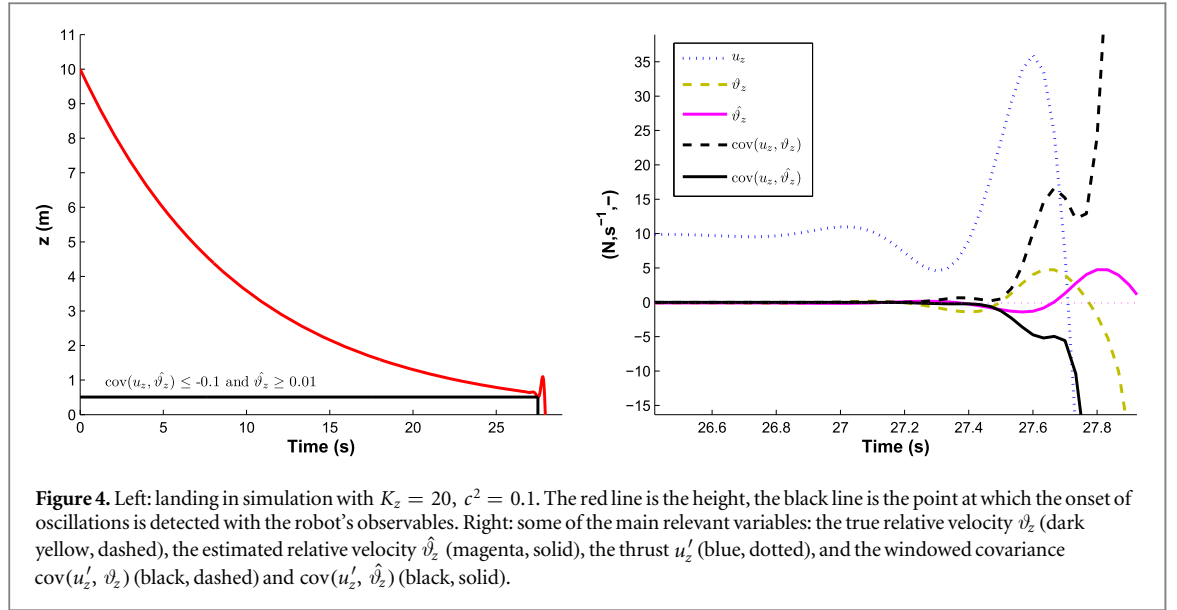
and a rather involved transfer function $H(w)$. The root locus plot of $H(w)$ is very similar to that of $G(w)$, also with a negative infinite zero. Following the same procedure as for the ‘vacuum’ model, setting $w = -1$ and equating the denominator with 0, gives:

$$K_z = \frac{(2p^2 + 2p^2 e^{pT})z^2}{(2e^{pT} - 2 - Tp - Tpe^{pT})v_z + (2pe^{pT} - 2p)z}. \quad (31)$$

Equation (32) includes many instances of p that depend on v_z and v_{wind} , and also contains a term in the denominator with v_z . This suggests that there is no fixed linear relation between K_z and z . However, closer inspection shows that the term $(2e^{pT} - 2 - Tp - Tpe^{pT}) \approx 0$. Rearranging terms, this leads to:

$$K_z = \frac{2p^2 + 2p^2 e^{pT}}{2pe^{pT} - 2p}z, \quad (32)$$

which still depends on p . It turns out that the fraction that is multiplied with z is almost identical to the fraction $\frac{2}{T}$. Indeed, solving equation (32) for different



variable settings gives almost identical results to equation (24).

3.4. Continuous system with drag and delay

The previous subsections discussed discretized models for which unstable K_z exists, and where the limit case can be analytically expressed as a function of z . In order to show that continuous systems encounter the same type of phenomenon, figure 3 shows the root locus plot of a continuous system with wind, a delay of $\Delta t = 0.03$ s and $c^2 = 0.01$. The delay introduces two zeros in the (unstable) right-hand plane of this continuous root locus plot. A few values of K_z are plotted, where the poles get an imaginary component and where they cross the imaginary axis (shown for $z = 10$ m and $z = 1$ m).

4. Stability-based distance estimation

In the previous section, it was established that the instability of a constant divergence approach system depends on the distance. If this instability can be detected by a robot or insect, it can be used for distance estimation!

4.1. Detection of self-induced oscillations

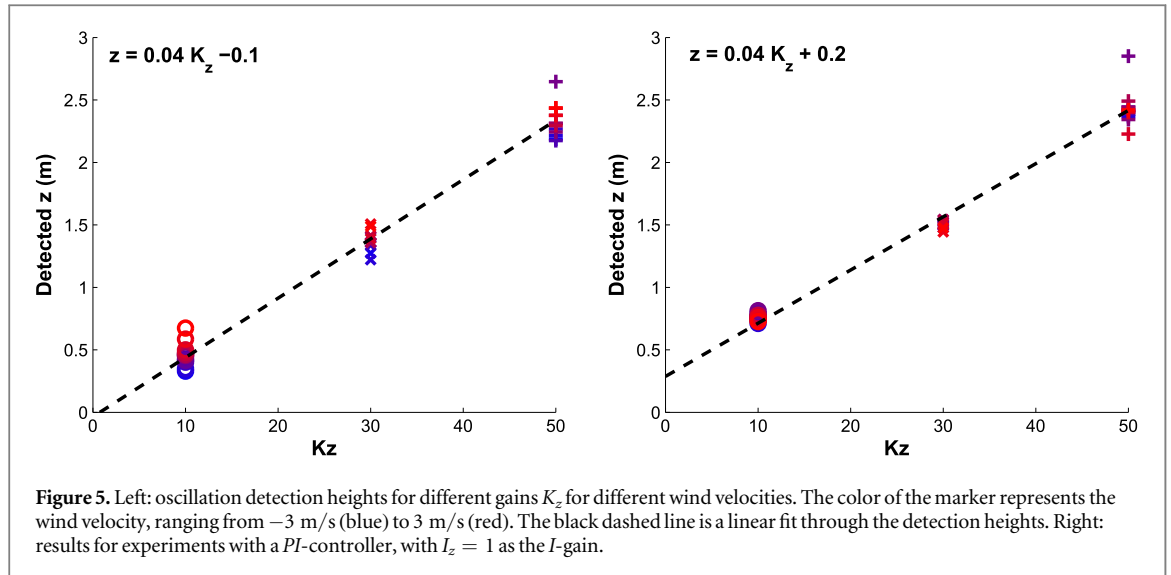
The instability discussed in section 3.2 is induced by the robot itself. Before the system gets unstable, it will start to show oscillations. This can be seen for instance in figure 3. When $K_z = 24.3$, the control system's poles will start to have an imaginary component and hence oscillate around the desired value from $z = 10$ m downward. In the field of aerospace, such self-induced oscillations are a well-known problem, and referred to as pilot-induced oscillations (PIO).

The two most important properties of self-induced oscillations are that (1) there is a phase shift between the observations and control inputs in the

order of 90° – 180° , and (2) the oscillations are of a significant magnitude. Let us have a look at a simulated landing to see how this phenomenon occurs for the studied constant divergence landings. For the simulations, a ZOH-model is assumed with a time step $T = 0.03$ s, $\frac{1}{2}\rho C_D A = 0.5$, a mass of 1 kg, and a delay of $\Delta t = 0.15$ s (as in section 3.4)². The left part of figure 4 shows a landing for $K_z = 20$, $z_0 = 10$, $v_{z0} = -1$, $c^2 = 0.1$, in wind still conditions ($v_{\text{wind}} = 0$). It shows that the landing goes smoothly, until the very end at ≈ 0.5 m at which point the robot's height starts to oscillate.

The right part of figure 4 shows the main relevant variables, zooming in at the few seconds before landing (until $t = 26$ s the variables do not vary much). The dark yellow dashed line is the ground-truth $\vartheta_z = v_z/z$, while the magenta solid line is its delayed version $\hat{\vartheta}_z$ as observed by the robot. The blue dotted line represents the thrust u_z' . Toward the end of the landing ϑ_z starts deviating significantly from its reference point -0.1 (indicated with the thin dotted magenta line). It even obtains positive values, which means that the robot goes up. This itself could be used for the detection of instability, were it not that a wind gust can also move the robot upward, causing a positive ϑ_z at a larger height. The problem here is that the effects of the thrust changes have an ever larger and quicker effect on the relative velocity when getting closer to the landing surface. This response will get so quick that the robot starts getting in resonance with its own delay, as testified by the phase shift between ϑ_z and $\hat{\vartheta}_z$. The robot starts thrusting up while going up and thrusting down while going down.

² Please note that while the theoretical analysis involved a linearization of the observation function, the simulation employs the nonlinear observation function (and hence can validate some of the conclusions from section 3). For the simulator code, please see www.bene-guido.eu.



The two properties of self-induced oscillations (phase shift and magnitude) can be captured with the *covariance* between the ground-truth ϑ_z and the thrust u'_z . The right part of figure 4 shows $\text{cov}(u'_z, \vartheta_z)$ as determined over a window of the last $W = 20$ time steps (black dashed line):

$$\text{cov}_t(u'_z, \vartheta_z) = \sum_{i=t-W+1}^t (u'_z(i) - \bar{u}'_z)(\vartheta_z(i) - \bar{\vartheta}_z). \quad (33)$$

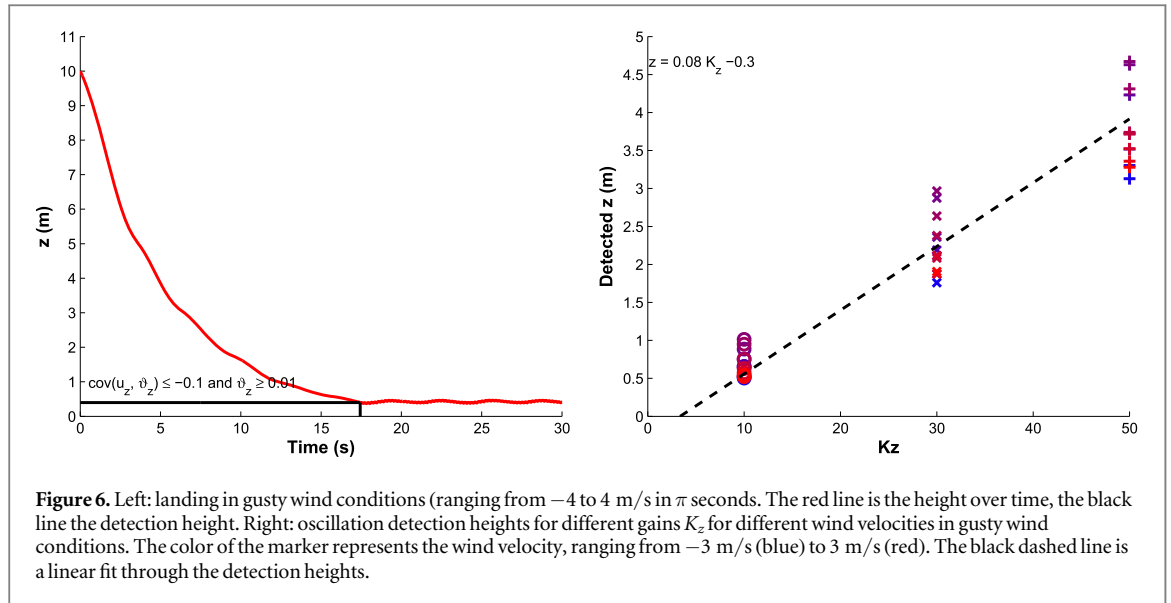
During the smooth part of the trajectory $\text{cov}(u'_z, \vartheta_z)$ is a small negative number: when the robot descends too fast (ϑ_z too negative) the robot thrusts more up, and vice versa. Towards the end of the landing the observable ϑ_z starts to change too quickly for the control system and $\text{cov}(u'_z, \vartheta_z)$ becomes positive: if the robot descends too fast (ϑ_z too negative) the robot will thrust *less* up, and thus it starts inducing oscillations.

How can robots detect self-induced oscillations without access to the ground truth ϑ_z ? There have been several investigations on the automatic, on board detection of self-induced oscillations [7, 10, 34, 38, 47]. A typical detection method involves a fast Fourier transform (FFT) of the observations [7, 38], looking for the specific frequency band related to the self-induced oscillations. In this article, in the interest of a computationally efficient and straightforward method, we will investigate the use of the $\text{cov}(u'_z, \hat{\vartheta}_z)$ as a proxy for $\text{cov}(u'_z, \vartheta_z)$. In figure 4, $\text{cov}(u'_z, \hat{\vartheta}_z)$ is shown with a black solid line. It is always negative, as $u'_z = K_z(\vartheta_z^* - \hat{\vartheta}_z)$, and hence mostly captures the magnitude of the oscillations. The $\text{cov}(u'_z, \hat{\vartheta}_z)$ can be used in conjunction with ϑ_z for detecting the onset of oscillations. The black line in the left part of figure 4 shows the first point during the landing at which $\text{cov}(u'_z, \hat{\vartheta}_z) \leq -0.1$ and $\hat{\vartheta}_z \geq 0.01$.

In order to test the hypothesis that the onset of oscillations is related to the magnitude of the gain and

the height of the robot, simulation runs were performed for gains $K_z \in \{10, 30, 50\}$ and for various types of wind $v_{\text{wind}} \in \{-3, -2.5, \dots, 2.5, 3\}$ m/s. Furthermore, $z_0 = 10$, $v_{z0} = -1$, $c^2 = 0.05$. The left part of figure 5 shows the results of this experiment, with the different K_z on the x -axis and the corresponding z at which self-induced oscillations are detected on the y -axis. Each K_z additionally is represented with a different marker, and the markers are colored according to the wind velocity, from blue (-3 m/s) to red (3 m/s). The black dashed line is a linear least-squares fit through the points, with parameters $z = 0.04K_z - 0.1$.

There are three main observations. First, as the linear fit shows, higher gains result in oscillations further away from the landing surface. Hence, a fixed gain will result in self-induced oscillations around a certain height. The detection of these oscillations can be used for instance for triggering landing responses (such as the leg extension by fruitflies). Second, a considerable wind difference between -3 to 3 m/s leads to a relatively small (for MAVs) difference of ≈ 0.50 m in the height at which self-induced oscillations occur. Third, the colors show that there is a slight correlation between the detection height and the wind velocity. Analyzing the landings with wind shows that for the higher wind velocities, the control system has more difficulties tracking ϑ_z^* , leading to a steady-state error. This suggests that introducing an integrator term could resolve this problem. The results of a PI -controller, with K_z as the P -gain and $I_z = 1$ as the I -gain, are shown in the right part of figure 4. Since the PI -controller can cancel the steady-state error, the reference ϑ_z^* is tracked much better. As expected, self-induced oscillations still occur, and the detection heights are much closer together. Moreover, inspection of the marker colors shows that there is no obvious relation anymore between the wind velocity and the detection height. In order to keep in line with the system that was analyzed theoretically, for the



remainder of the article we will focus on the P -controller setup with gain K_z .

4.2. Wind gusts and actuator efficiency

In the previous subsection, the feasibility of detecting self-induced oscillations for distance estimation was shown. However, two factors were not modeled that could have a significant influence on the practicality of the approach.

First, during a typical landing outdoors the wind is not constant. The wind can vary and sudden wind gusts can occur. In the simulation, wind gusts are added to the simulation in the form of a sine function:

$$v_{\text{gust}} = W \sin(at), \quad (34)$$

where W determines the magnitude of the gusts and a the period.

Second, it is unlikely that a robot's command signal (and hence reference copy) is equal to u'_z . A command signal u''_z (in any unit, e.g. the commanded RPM of a robot's propellers or the flapping frequency of an insect) is related to u'_z (in Newton) with an actuator effectiveness function: $u'_z = f(u''_z)$. In flying robots such as rotorcraft or flapping wing vehicles the function f , in turn, depends on the air flow. Here, a rotorcraft is assumed and the relation between actuator effectiveness and air flow is modeled according to findings on propellers in [44]. Specifically, the following formula was used:

$$u'_z \leftarrow \max\{u'_z - bv_{\text{air}}u'_z - cv_{\text{air}}, 0\}, \quad (35)$$

so that both the offset and slope of the efficiency change over different air velocities.

The left part of figure 6 shows a landing with a rather extreme scenario, in which the average wind velocity is 0 m/s, $W = 4$, $a = 1$, $b = 0.5$, and $c = 0.5$. This means that per every π seconds, the wind varies from -4 to 4 m/s! The red line in the figure is the

height over time, the black line indicates the height at which self-induced oscillations are detected. The right part shows the results over many landings with the same settings but different wind velocities, from -3 m/s (blue) to 3 m/s (red), to which the gusts are added during each landing. The black dashed line is a linear fit through all the detection heights for all gains K_z .

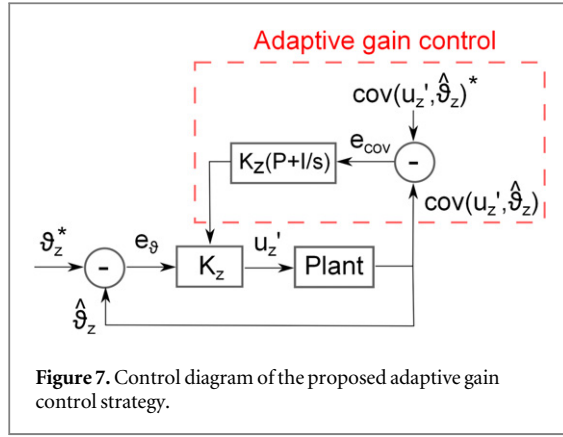
The results show that despite the very gusty conditions and varying actuator efficiency, there is still a positive linear relation between z and K_z . The uncertainty is slightly higher though than for a constant wind and actuator efficiency, especially for larger gains/heights. It is important to emphasize that the actuator efficiency function f does not have to be known. Assuming a linear actuator efficiency function and using that the fit should pass through $(K_z, z) = (0, 0)$, a single landing with a fixed gain in principle suffices to 'calibrate' the linear relation between z and K_z .

5. Adaptive gain control

5.1. Distance estimation in hover

In principle, the detection of self-induced oscillations could be sufficient for flying robots (or insects) to determine when they are close to their landing target. This allows for the triggering of a final landing procedure and hence is behaviorally very relevant. However, it would be extremely useful if the robot could determine height at any distance to the landing surface.

Surprisingly, the proposed stability-based strategy for height estimation does not require the robot to land! Equation (25) (derived for the vacuum condition) does not depend on v_z , nor on c^2 . This suggests a strategy for determining the height around hover: the



robot can set $\vartheta^* = 0$ and change its gain K_z so that the control loop starts exhibiting small self-induced oscillations. The gain at such a point is then a stand-in for the height.

Specifically, a control law can regulate $\text{cov}(u_z', \hat{v}_z)$ by continuously adapting the gain K_z . This leads to the following adaptive gain control setup. An inner loop uses $u_z = K_z(\vartheta_z^* - \hat{v}_z)$, while an additional outer loop controls K_z :

$$K_z(t) = K_z'(t) - (P K_z'(t))e_{\text{cov}}(t), \quad (36)$$

$$K_z'(t) \leftarrow K_z'(t - T) - (I K_z'(t))e_{\text{cov}}(t - T), \quad (37)$$

$$e_{\text{cov}}(t) = \text{cov}(u_z', \hat{v}_z)^* - \text{cov}(u_z', \hat{v}_z)(t), \quad (38)$$

where $P, I \in [0, 1]$ are a proportional and integral gain for the outer loop control, both relative to the current K_z' . The reason for this is that larger heights involve higher speeds, so that K_z should be changed more quickly for larger K_z' than for smaller K_z' . The adaptive gain control is represented as a control diagram in figure 7.

This strategy has been tested in simulation for $z \in \{2, 4, \dots, 10\}$ and $v_{\text{wind}} \in \{-1, 0, 1\}$. The simulation stops when $|e_{\text{cov}}| < 0.005$. Figure 8 shows for each simulation run the last gain and height with a marker. The markers are again color-coded for the wind from blue (-1 m/s) to red (1 m/s). The black dashed line shows a linear least squares fit.

The main result is that this method indeed gives an approximately linear relationship between z and K_z , with linear fit $z = 0.07K_z + 0.1$. Although the wind influences the hover altitude (since it is not accounted for in the initial thrust) it again hardly influences the results.

5.2. Distance estimation during landing

Is there a possibility of landing and continuously estimating the distance to the surface? Figure 8 suggests that there is. If the robot descends while keeping $\text{cov}(u_z', \hat{v}_z)$ at a fixed negative value, then the gain K_z will directly represent the height z during the landing. This strategy of ‘landing on the edge of oscillation’ can

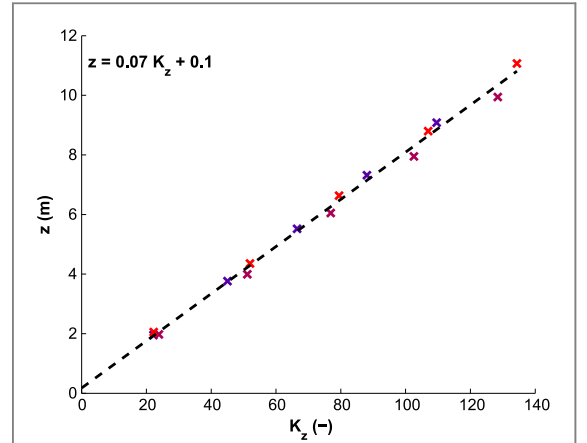


Figure 8. Results of simulated hover tests. The robot is placed at different heights with $K_z = 10$ with $\vartheta^* = 0$. The gain is then adapted with $\text{cov}(u_z', \hat{v}_z)^* = -0.05$. The x-markers show the height z and K_z at the first time instant in which $\text{cov}(u_z', \hat{v}_z) \leq -0.05$. The markers are color-coded for the wind (blue = -3 , red = 3). The black dashed line is a linear fit.

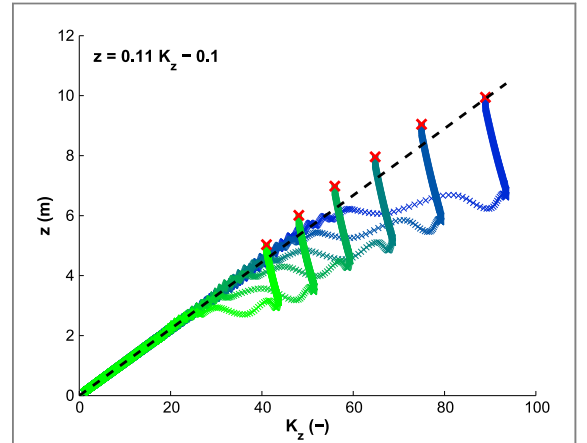
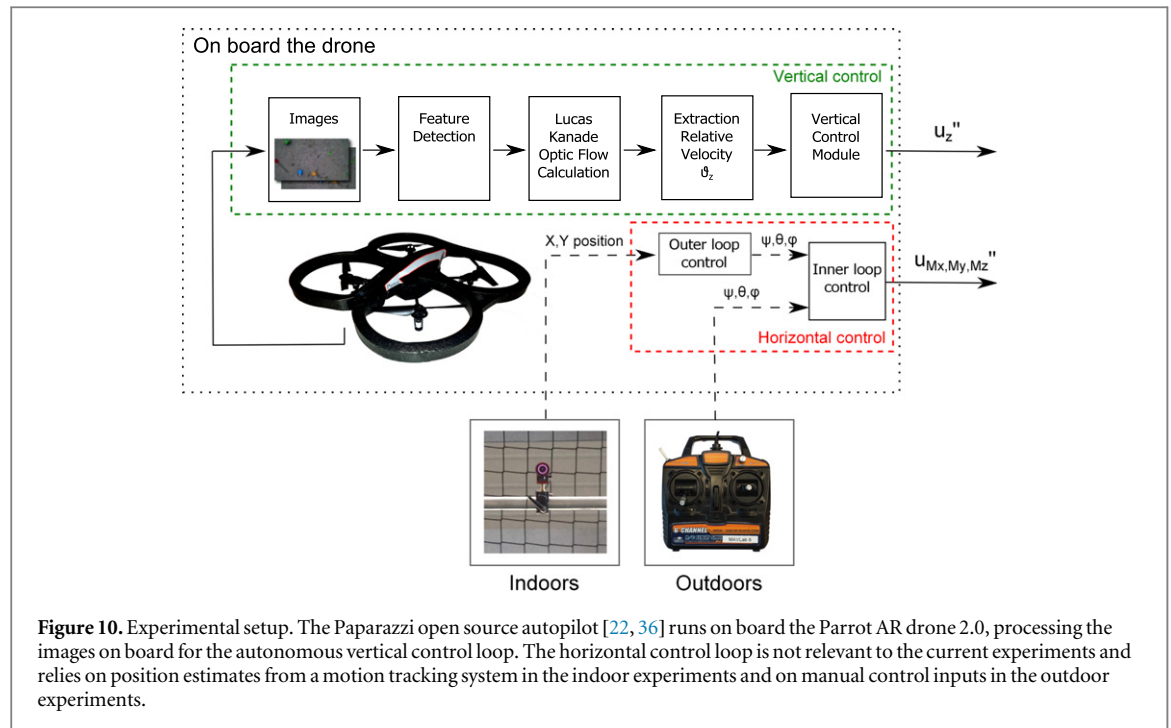


Figure 9. Results of simulated tests landing on the edge of oscillation. The robot is placed at different heights with $K_z = 50$. Then, it starts adaptive gain control with $\text{cov}(u_z', \hat{v}_z)^* = -0.05$. First $\vartheta^* = 0$, and when for the first time $\text{cov}(u_z', \hat{v}_z) \leq -0.05$, the relative velocity set point is changed to $\vartheta^* = -0.05$.

use the exact same adaptive gain control as explained above, but then with $c^2 > 0$.

Simulations of this strategy have been performed for $z \in \{5, 6, \dots, 10\}$. The implementation of the strategy starts with a hover maneuver ($c^2 = 0$) with $K_z = 50$, $I = 0.005$, and $P = 0.15$. If in hover the condition $\text{cov}(u_z', \hat{v}_z)^* = -0.05$ is met, the landing is commenced. During landing $c^2 = 0.05$, $\text{cov}(u_z', \hat{v}_z)^* = -0.05$.

Figure 9 shows the K_z versus z during the landing phase. The markers are color-coded for the starting height z_0 in the landing phase, from blue (10 m) to green (5 m). Each starting point is indicated with a red cross. After switching from $c^2 = 0$ to $c^2 = 0.05$, it takes some time for the outer loop to compensate, which causes K_z to vary at the start of the landing



phase. After that, all K_z are linearly related to z , with linear fit $z = 0.11K_z - 0.1$.

6. Real-world experiments

In this section, experiments will be performed with a Parrot AR drone in order to verify the main findings from the theoretical analysis and simulations in the previous sections.

6.1. Experimental setup

The experimental setup is shown in figure 10. A Parrot AR drone 2.0 is used, running the *Paparazzi* open source autopilot software on board [22, 36]³. This means that the Paparazzi autopilot is in full control of the AR drone, processing the sensors on board. The AR drone is equipped with an IMU (accelerometers, gyros, magnetometer, pressure sensor), and a downward pointing camera and sonar. For the purpose of the experiments, the vertical control loop implements the fixed and adaptive gain controls explained in the previous sections, and hence only uses the relative velocity estimates coming from the vision process, which is further explained below. The sonar is logged for analysis purposes. The horizontal position is stabilized as much as possible. In the indoor experiments, an Optitrack motion tracking system measures the drone's X - and Y -position at 120 Hz with millimeter precision. Paparazzi then performs the outer loop control on the basis of these measurements. In the outdoor experiments, where the drone is subject to much larger disturbances from the wind, a human

pilot attempts to keep the drone in the same horizontal position by means of manual control.

The vision process is designed as follows. The images are processed with a rather standard vision pipeline available in Paparazzi. Maximally 40 corners are detected with the method in [40], and these corners are tracked to the next image with the Lucas–Kanade optical flow algorithm [31]. The optical flow divergence is estimated by means of the distances between tracked corners in two subsequent images. From all possible pairs of optical flow vectors, 50 pairs are sampled with replacement. Per pair of optical flow vectors, the image distance in the previous image $d_{t-\Delta t}$ and the image distance in the current image d_t (see figure 11) are used for a single estimate i of the divergence:

$$D_i = d_{t-\Delta t} / (d_t - d_{t-\Delta t}). \quad (39)$$

A global divergence estimate \hat{D} is obtained by taking the median of estimates D_i , $i \in \{1, 2, \dots, 50\}$. With the help of the frame rate (which is on average 25 Hz) and the camera's field of view (47.5° in the horizontal direction), this estimated divergence is transformed to the estimate of the \hat{v}_z expressed in s^{-1} .

The estimated relative velocity \hat{v}_z is low-pass filtered over time and then used in a control loop $u'_z = K_z(\hat{v}_z^* - \hat{v}_z)$, where u'_z is the vertical thrust command. Please note that in the real-world experiments, the thrust command is bounded to the interval $u'_z \in [0.6M, 0.9M]$, where M is the maximal thrust that can be commanded to the propellers. The program allows both fixed gain control and adaptive gain control as in section 5.2. Experiments were performed in an indoor and an outdoor environment, shown in figure 12. The main parameters used in the experiments have been determined empirically for the used

³ See www.bene-guido.eu for the most current link to the code.

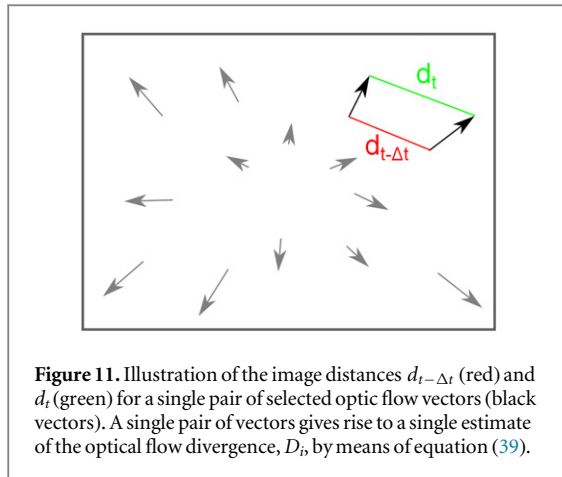


Figure 11. Illustration of the image distances $d_{t-\Delta t}$ (red) and d_t (green) for a single pair of selected optic flow vectors (black vectors). A single pair of vectors gives rise to a single estimate of the optical flow divergence, D_t , by means of equation (39).

AR drone 2.0 and will be mentioned along with the relevant experiment.

6.2. Indoor experiments

First, it is verified that a fixed-gain constant divergence landing results in instability at different heights. The results of this test for $K_z \in \{0.5, 1, 2\}$ (green, blue, and red lines) and $c^2 = 0.1$ are shown in figure 13. The left plot shows the height over time for nine such landings. Dotted lines indicate the first point at which $\text{cov}(u'_z, \vartheta'_z) \leq -1.0$, set to trigger the drone's automatic landing procedure. This value is larger than in simulation to prevent too-early landing triggering. The reason is that the real drone does not only have a delay in the observations, but also noise and it is subject to larger disturbances and uncertainty in the actuation. The main observation from the figure is that also for the real robot, higher gains lead to an instability at larger heights. The right part of figure 13 shows a marker for each point (K_z, z) at which self-induced oscillations are detected, and a corresponding linear fit, with parameters $z = 1.01K_z - 0.2$. The mean absolute error when using this function for height estimation is 0.11 m.

The second finding to verify is the determination of height around hover. For the real robot, $\text{cov}(u'_z, \vartheta'_z)^* = -0.025$ is employed, with $I = 0.25$ and $P = 10$. During the hover experiment, the robot has been sent to different heights, subsequently activating the adaptive gain control. In order to get an impression of what the adaptive control does, let us first zoom in on the relevant variables during a short part of the flight, shown in figure 14. In this part of the flight, the drone has just ascended to ~ 2 m (see top left plot), while it has an initial gain of $K_z = 1.0$ (bottom left). The drone starts to regulate its relative velocity to $\vartheta_z^* = 0$ (top right), and initially there are no self-induced oscillations as witnessed by the $\text{cov}(u'_z, \vartheta'_z) \approx 0$ (bottom right). The low covariance makes the gain K'_z increase over time. This gradually causes ϑ_z to show ever larger oscillations, which leads

to more negative $\text{cov}(u'_z, \vartheta'_z)$. Around $t = 114.5$ s, the covariance reaches and overshoots its set point, and the gain K'_z starts stabilizing around 1.6 in order to keep the covariance around -0.025 . The question now is at what values K'_z will stabilize at other heights, and whether there is indeed a linear relationship between z and K_z .

The middle row of figure 15 shows the result of having the drone hover at different heights with $\text{cov}(u'_z, \vartheta'_z)^* = -0.025$. The left plot shows a short time period during which the drone is continuously adjusting its gain (K_z and K'_z shown with the solid and dashed green lines, respectively), while attempting to keep the same height (z is represented by the red line). The right plot summarizes the results of five of such periods. The small blue markers represent the instantaneous (K_z, z) -pairs, while each large red marker is the average (\bar{K}_z, \bar{z}) over each period. The dashed line is a linear least squares fit through the red markers. Although more noisy than in simulation, the relation between z and K_z seems to be roughly linear, as predicted by the theoretical derivation. When using the linear fit $\hat{z} = 0.93K_z + 0.3$ as a height estimation function, the median absolute error of all observations is $|z - \hat{z}| = 0.15$ m, while the mean absolute error is 0.26 m.

The third finding to verify is the possibility to land on the edge of oscillation. The AR drone is sent to a height of 3.5 m. Then, the adaptive gain control is started, at first with $c^2 = 0$. As soon as $|e_{\text{cov}}| < 0.005$, the landing phase starts. The bottom row of figure 15 shows the results for this test. The bottom left plot shows K_z , K'_z , and z over time during a single landing on the edge of oscillation. The right plot shows the points (z, K_z) of this and two other landings, where different experiments have different markers and colors. The dashed line is a linear fit through all points. Although the results are quite noisy, again a roughly linear relation appears, with as linear fit $z = 0.68K_z + 0.5$. In this case, the median absolute error of all observations is $|z - \hat{z}| = 0.24$ m, while the mean absolute error is 0.29 m.

6.3. Outdoor experiments

Outdoor experiments have been performed to confirm that the method also works in an outdoor environment with wind and wind gusts. The right part of figure 12 shows the outdoor experimental setup. The ground surface consisted of small stones, which already provide ample texture. Some beach toys were added to the scene to also ensure visible texture at larger heights. The bottom left part of figure 16 shows the summary of the results from the hover experiment. Just as indoors, there is a roughly linear relationship between the height z as measured by the sonar and the gain K_z as determined with the adaptive gain control. The median and mean absolute errors in height of the fit are both 0.31 m. The bottom right part of figure 16 shows the results from three landings on the edge of



Figure 12. Left: indoor test environment. Right: outdoor test environment.

oscillation, again showing higher gains at larger heights. The median absolute error in height of the fit is 0.32 m, the mean is 0.40 m.

Videos of some of the experiments can be seen at ⁴.

7. Discussion

7.1. Flying insects

The presented strategy forms a novel hypothesis on how flying insects such as fruitflies and honeybees can estimate distances, viz. by means of the detection of self-induced oscillations. In [6] a different, thrust-based strategy for distance estimation was introduced, and its implications for landing insects discussed. In part, the implications of these strategies overlap. They both explain how flying insects can trigger landing responses at a particular given distance, such as fruitflies that extend their legs just before landing (e.g. [5]). However, there is a significant difference in the strategies, which has its consequences for the explanation of flying insects' behavior. A few notable differences in this respect will be highlighted below.

A first difference is in the possible explanation of the hover phase honeybees exhibit just before touchdown [13]. The hover phase could in principle be explained as a maneuver that is triggered by a distance measurement, for instance by means of the thrust-based strategy of [6]. However, the proposed strategy suggests instead that the hover mode is an intrinsic property of optical flow control and is part of the distance estimation itself. As the honeybee gets closer to the surface, its control gain will start becoming too high. This will result in self-induced oscillations and lead to a situation of zero divergence, i.e. hover. In this light, it is interesting to remark the characteristic little 'bump' in the trajectory in figure 4. Similar bumps can be observed in figure 4 in [13] (for almost straight landings), and in figure 5 in [42] (for grazing landings).

A second difference is that the stability-based strategy explicitly allows for the observation that a visual stimulus itself can trigger the landing responses, even

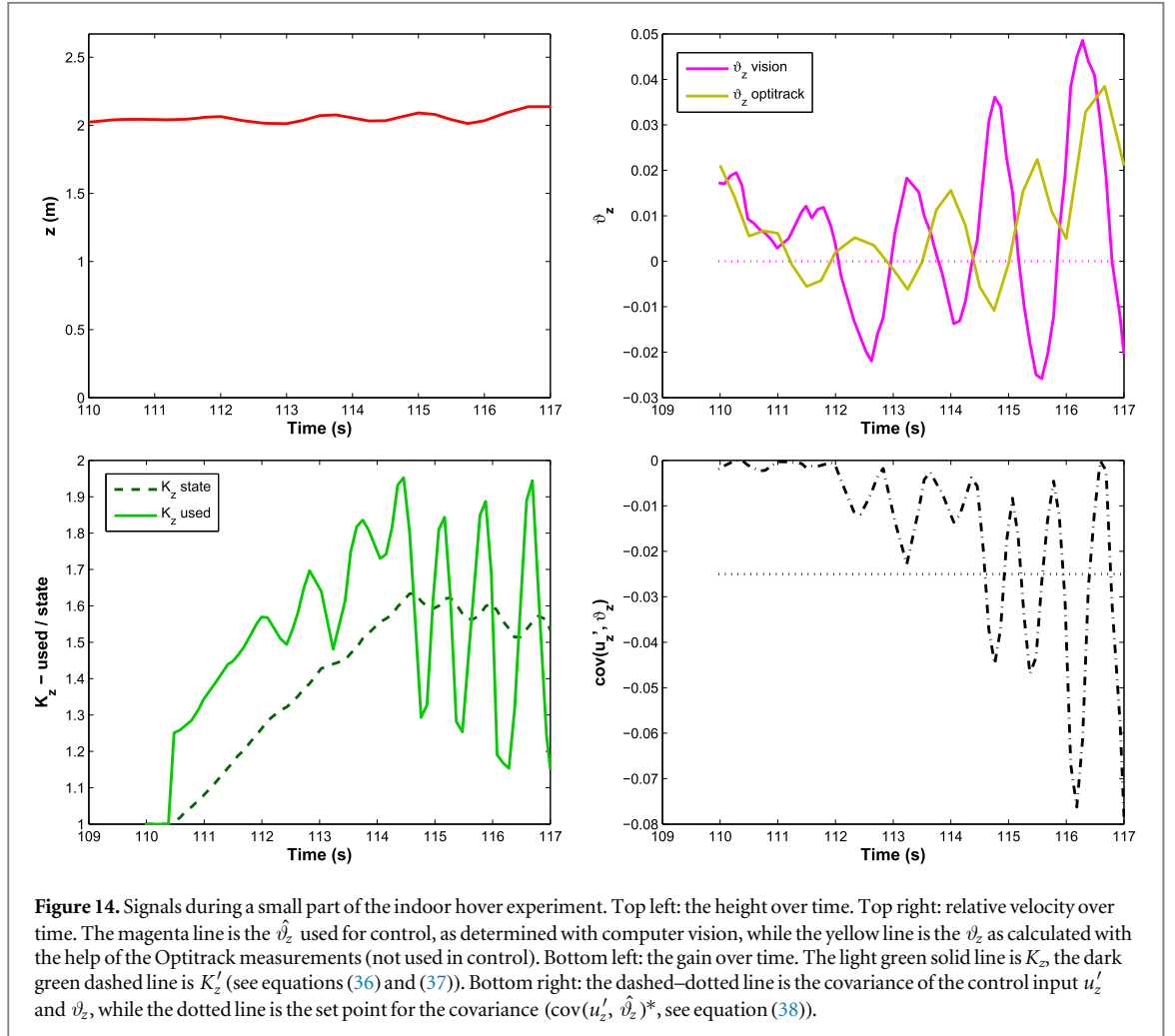
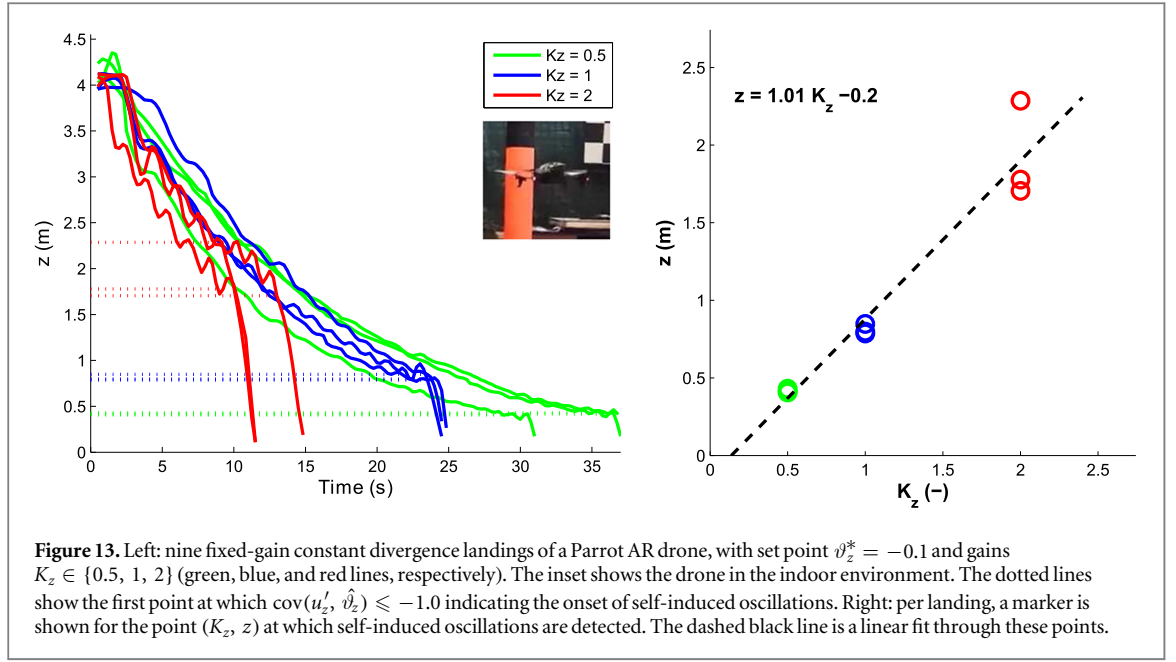
for tethered insects [4, 43]. The reason for this is that the strategy depends on the control stability and hence the system bandwidth. A standard way to measure the system bandwidth is to make a Bode plot that maps the system's frequency response. At too-high frequencies, the phase shift between the quickly changing observation and the subsequent control input will become larger and the magnitude of the control input lower. If the magnitude of the triggered control input is large enough, a purely visual stimulus may be detected as a self-induced oscillation.

7.2. Flying robots

Concerning flying robots, the novel proposed distance estimation strategy provides a novel way to perceive distances with a single vision sensor. This may turn out to be essential for tiny flying robots such as the Robobee [33], for which any type of sensor is a significant payload [20, 23, 39]. For slightly larger drones such as 25 g 'pocket drones' (e.g. [11]), the finding is also immediately relevant. For instance, currently the lightest, fully autonomously flying robot is the 20 g DelFly Explorer [46], which uses a 4 g stereo vision system with onboard processing to avoid obstacles in its environment. The proposed stability-based strategy to distance estimation may provide a bigger potential to even lighter, single-camera solutions, also for obstacle avoidance. In addition, it can provide scale to monocular navigation methods such as visual odometry [12, 17]. Even for MAVs on the order of ~ 1 kg or larger, the method may still be useful. It allows us to estimate potentially large distances without compromising the MAVs' payload capability.

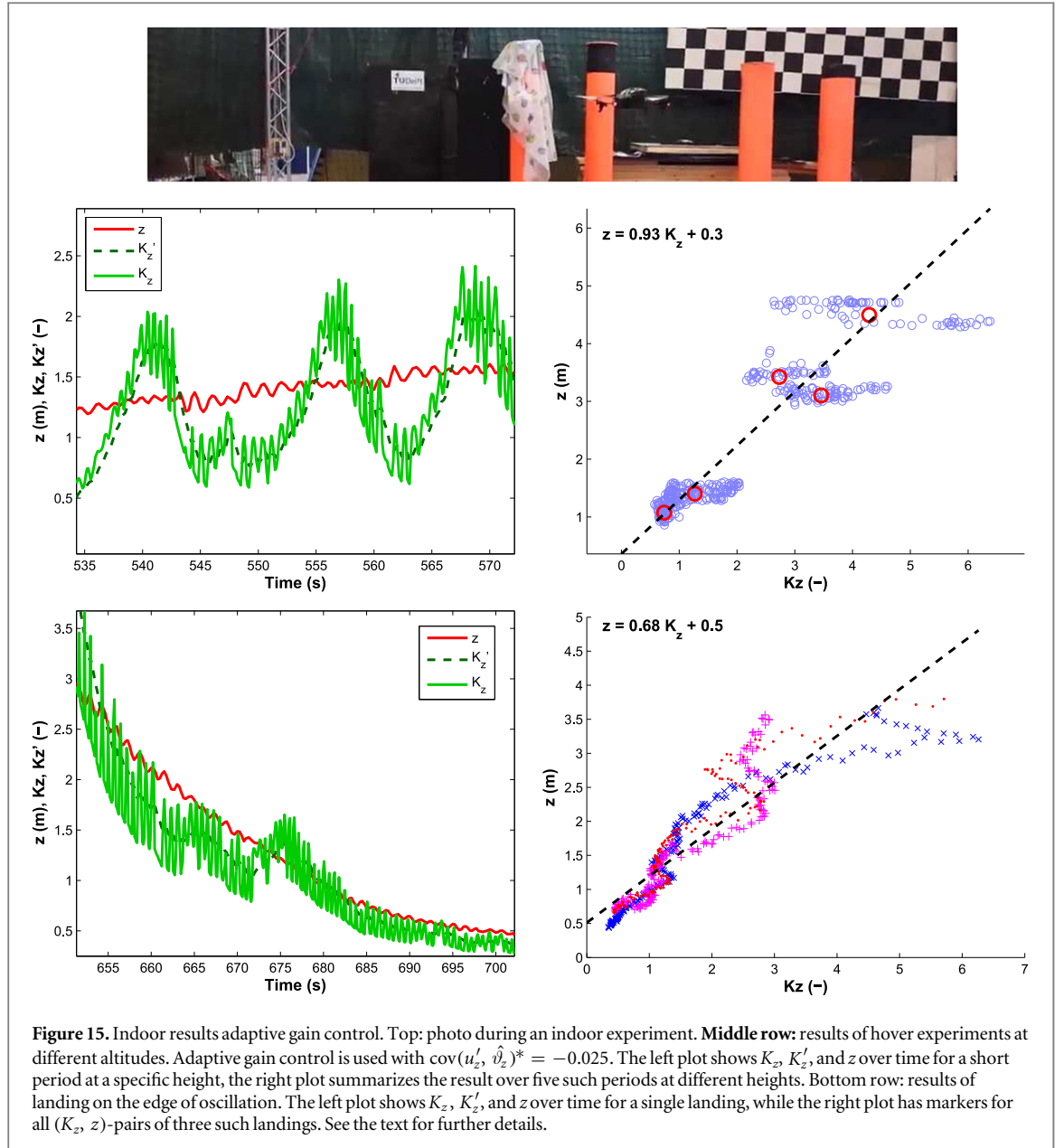
However, in order for stability-based distance estimation to live up to its potential, some issues need to be further investigated. The experiments have shown the strategy to work in an onboard processing scheme with a Parrot AR drone 2.0 with a median height estimation error in the order of ~ 0.3 m. As a proof of concept this suffices, but it is desirable to improve the accuracy of the measurement. This seems quite possible by further improving both the inner loop divergence control (which now only involved a P -gain) and

⁴ www.youtube.com/playlist?list=PL_KSX9GOn2P96xb8JRS0jLFnLBK0Pj5Y



the outer loop adaptive control. Moreover, other, more reliable ways of detecting self-induced oscillations may be created. Such improvements could not only lead to more accurate distance estimates, but also

to smoother divergence control. If high-performance, but smooth landing trajectories are the goal, the gain should always be slightly below the critical gain value leading to self-induced oscillations.



8. Conclusion

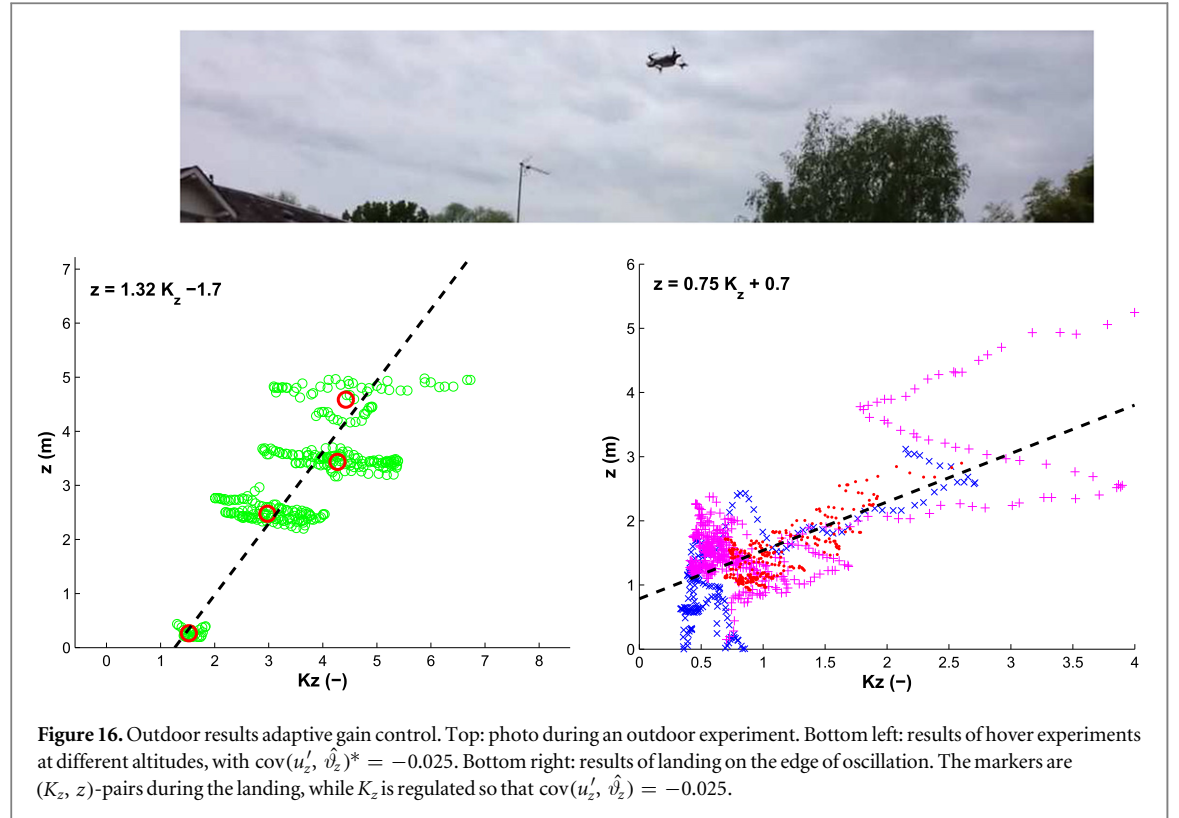
In this paper, a stability-based strategy has been proposed with which a robot can estimate distances only on the basis of efference copies and optical flow maneuvers. Theoretical analysis of linearized models has shown that there is a linear relation between the fixed gain and the height at which instability arises. This analysis has been verified in simulation (including non-linearities and disturbing factors such as wind gusts) and on a real robotic platform. It has been demonstrated that self-induced oscillations can be detected by the robot and used to (1) trigger a final landing procedure, (2) determine the height in hover, and (3) continuously determine the height during a constant divergence landing.

Acknowledgments

I would like to thank Christophe De Wagter, Hann Woei Ho, and Tobias Seidl for the interesting discussions and comments on earlier versions of the manuscript.

Appendix A. Influence of wind on thrust-based distance estimation

In the introduction, it is mentioned that the ‘thrust’-based strategy of [6] does not take into account external accelerations. In this appendix, some insight is provided into the effects on the distance estimates when introducing external accelerations by wind.



Let us first re-arrange equation (11):

$$z = \frac{a_z}{\vartheta_z^2 + \dot{\vartheta}_z}. \quad (40)$$

This expresses z in terms of ‘observables’ (ϑ_z and its time derivative $\dot{\vartheta}_z$), and the vertical acceleration a_z . Adding an accelerometer to the robot will allow for estimation of z (e.g. [25]). Besides the addition of the accelerometer, this method also requires the time derivative of ϑ_z , which typically induces a lot of noise in the observation.

In [6], the assumption is made that $a_z = u_z$, so that equation (40) becomes:

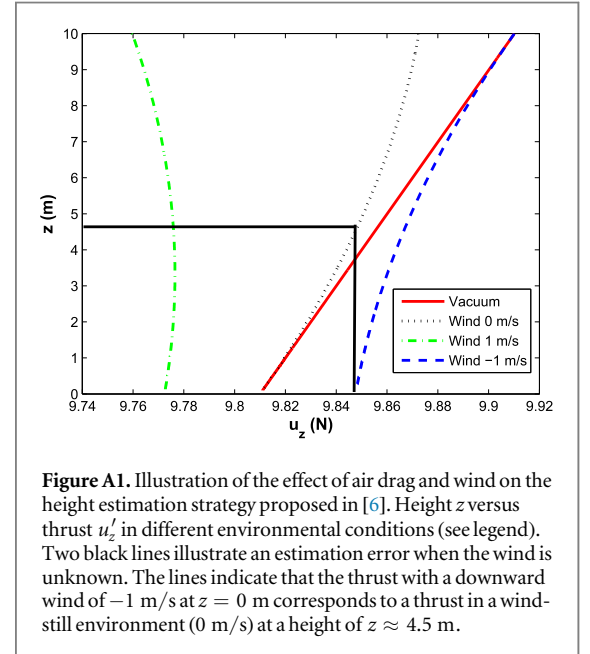
$$z = \frac{u_z}{\vartheta_z^2 + \dot{\vartheta}_z}. \quad (41)$$

In [6] it has been investigated what would happen if a perfectly constant divergence landing is performed, $\dot{\vartheta}_z = 0$ and $\vartheta_z = -c^2$ (c^2 to indicate that ϑ_z has a negative set point):

$$z = \frac{u_z}{c^4}. \quad (42)$$

Equation (42) shows that during a perfect constant divergence landing, the thrust u_z is a scaled version of the height. Hence, it can be used as a stand-in for the height. For this reason, the method is referred to in this article as a ‘thrust-based’ method.

It is informative to study the effect of drag and wind on the required u'_z , when the robot follows a perfect constant divergence landing. In such a landing, $\vartheta_z = -c^2$ at every point by definition. The acceleration resulting from drag and the required u'_z are



calculated to match. Figure A1 shows the value of u'_z for a robot with assumed values of $m = 1$ kg, $\rho = 1.204$, and $g = 9.81$ m/s². Furthermore, $C_D = 0.25$, and $A = 0.25$, which are rather conservative values representing for instance a small quad rotor. The figure shows the thrust u'_z versus height z in different environmental conditions: vacuum (red solid line), wind still, i.e., $v_{\text{wind}} = 0$ m/s (black dotted), in a downward wind of $v_{\text{wind}} = -1$ m/s (blue

dashed), and in an upward wind of $v_{\text{wind}} = 1 \text{ m/s}$ (green, dashed-dotted).

Three main observations can be made from figure A1: (1) the red solid line in a vacuum indeed indicates the derived linear relationship between u_z and z , (2) in a wind-still environment, this relationship is non-linear but invertible, and hence still as useful, and (3) adding a modest wind speed to the equation already makes significant differences to u'_z . Two black lines indicate that the thrust with a downward wind of $-1 \text{ at } z = 0 \text{ m}$ corresponds to a thrust in a wind-still environment at a height of $z \approx 4.5 \text{ m}$. The curve for a wind of 1 m/s does not even match that in a wind-still environment. Hence, even with a perfect constant divergence landing these differences distort the relationship between u_z and z .

So, in order to retrieve the right u_z -curve (see figure A1), the wind will have to be measured. Flying insects may be able to measure the air speed with their hairs [19], while robots can be equipped with an anemometer, as in [15]. Although measuring the air speed and deducing wind velocity is a possibility, this does not lead to an easy solution of the distance estimation problem. First and foremost, controlling a constant divergence landing in a real system means dealing with sensing and actuation delays, visual inaccuracies, and changing external factors such as wind gusts. Any real system will thus deviate from the perfect landing profile, leading to the necessity of command signal variation and hence a varying u'_z . Indeed, this effect can be seen in the raw \hat{z} estimates in [6] for the camera-system mounted on a rail. In [6] they tackled this issue with a robust estimation scheme that integrates information during the entire approach. The problems are much worse though for a freely flying system, which is also subject to wind.

Appendix B. Generalization to other movement directions

In section 4 it was shown that given a gain K_z , the stability of the vertical control loop with ϑ_z depends on the height z . In this appendix the case is studied in which the camera still looks down in the z -direction, but the movement is in the horizontal x -direction. The control is thus based on the relative velocity $\vartheta_x = \frac{v_x}{z}$. It will first be shown that the finding also generalizes to such horizontal optical flow control (section B.1). The stability-based method then does not require vertical motion. Subsequently, a thrust-based method is investigated for horizontal motion (section B.2). The thrust-based method always requires both horizontal and vertical motion.

B.1. Stability-based method

Let us start from the observation:

$$\gamma = \vartheta_x = \frac{v_x}{z}, \quad (43)$$

Since both the horizontal and vertical axes are involved, the state space model will have four variables. Linearizing the state space model gives:

$$\Delta y(t) = \begin{bmatrix} -\frac{v_x}{z^2} & 0 & 0 & \frac{1}{z} \end{bmatrix} \times \begin{bmatrix} \Delta z(t) & \Delta v_z(t) & \Delta x(t) & \Delta v_x(t) \end{bmatrix}^T, \quad (44)$$

so that the state space model matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ C = \begin{bmatrix} -\frac{v_x}{z^2} & 0 & 0 & \frac{1}{z} \end{bmatrix} \quad D = [0]. \quad (45)$$

Again, the discretized system is studied, which has the following state space model matrices corresponding to the continuous ones in equation (45):

$$\Phi = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} 0 \\ 0 \\ \frac{T^2}{2} \\ T \end{bmatrix} \\ C = \begin{bmatrix} -\frac{v_z}{z^2} & 0 & 0 & \frac{1}{z} \end{bmatrix} \quad D = [0], \quad (46)$$

where T is the discrete time step in seconds. The transfer function of the open loop system can be determined to be:

$$G(w) = C(wI - \Phi)^{-1}\Gamma, \quad (47)$$

$$= \frac{T}{z(w-1)}, \quad (48)$$

where we use w as the Z -transform variable, since z already represents height. The feedback transfer function is:

$$G(w) = \frac{K_x T}{z(w-1) \left(\frac{K_x T}{z(w-1)} + 1 \right)}. \quad (49)$$

Equation (49) shows that, given a gain K_x and time step T , the dynamics and stability of the system depend on the height z .

Setting $w = -1$ in the denominator and equating it with 0 gives:

$$-2z \left(\frac{K_x T}{-2z} + 1 \right) = 0, \quad (50)$$

which leads to an equation that expresses the height at which the control system will become unstable in terms of the unstable gain value K_x :

$$z = \frac{1}{2} K_x T, \quad (51)$$

which is exactly the same formula as for the vertical movements studied in section 4.

B.2. Thrust-based method

It is instructive to also look at what a possible thrust-based method would look like for the horizontal motion direction. For this, let us again start from the observation ϑ_x , which is differentiated over time:

$$\dot{\vartheta}_x = \frac{a_x}{z} - \frac{v_x v_z}{z^2} = \frac{a_x}{z} - \vartheta_x \vartheta_z, \quad (52)$$

leading to:

$$z = \frac{a_x}{(\dot{\vartheta}_x + \vartheta_x \vartheta_z)}. \quad (53)$$

Equation (53) shows that the height can be estimated with the help of a horizontal accelerometer measurement and the observables ϑ_z , ϑ_x , and the time derivative $\dot{\vartheta}_x$ (as was done in [25]).

If the relative velocity ϑ_x (sometimes referred to as ‘ventral flow’) is kept constant, then $\dot{\vartheta}_x = 0$, and:

$$z = \frac{a_x}{\vartheta_x \vartheta_z}, \quad (54)$$

where in a vacuum environment, a_x can be assumed to be u_x . A constant ventral flow and divergence landing would then again lead to a way to estimate z , as u_x is known and ϑ_x and ϑ_z can be assumed equal to constants.

Interestingly, also this horizontal motion case does not allow for distance estimation while staying at the same height ($\vartheta_z = 0$). The reason for this can be seen when rearranging equation (54):

$$a_x = z(\vartheta_x \vartheta_z), \quad (55)$$

which shows that $\vartheta_z = 0$ implies $a_x = 0$. Indeed, if $\vartheta_z = 0$, ϑ_x can only be constant if there is no horizontal acceleration.

References

- [1] Alkowitz M T, Becerra V M and Holderbaum W 2015 Bioinspired autonomous visual vertical control of a quadrotor unmanned aerial vehicle *J. Guid. Control Dyn.* **38** 249–62
- [2] Asl H J, Oriolo G and Bolandi H 2014 An adaptive scheme for image-based visual servoing of an underactuated UAV *Int. J. Rob. Autom.* **29** 92–104
- [3] Baird E, Boeddeker N, Ibbotson M R and Srinivasan M V 2013 A universal strategy for visually guided landing *Proc. Natl Acad. Sci. USA* **110** 18686–91
- [4] Borst A and Bahde S 1986 What kind of movement detector is triggering the landing response of the housefly? *Biol. Cybern.* **55** 59–69
- [5] Van Breugel F and Dickinson M H 2012 The visual control of landing and obstacle avoidance in the fruit fly *Drosophila melanogaster* *J. Exp. Biol.* **215** 178398
- [6] van Breugel F, Morgansen K and Dickinson M H 2014 Monocular distance estimation from optic flow during active landing maneuvers *Bioinspir. Biomim.* **9** 025002
- [7] Chowdhary G V, Srinivasan S and Johnson E N 2011 Frequency domain method for real-time detection of oscillations., *Journal of Aerospace Computing J. Aerosp. Comput. Inf. Commun.* **8** 42–52
- [8] Corke P I and Good M C 1992 Dynamic effects in high-performance visual servoing *Proc. IEEE Int. Conf. on Robotics and Automation* 1838–43 IEEE
- [9] de Croon G C H E, Alazard D and Izzo D 2015 Controlling spacecraft landings with constantly and exponentially decreasing time-to-contact *IEEE Trans. Aerosp. Electron. Syst.* **1241**–52
- [10] Dimogianopoulos D G, Hios J D and Fassois S D 2005 On-board statistical detection and control of anomalous pilot-aircraft interactions *Proc. of the 16th IFAC World Congress*
- [11] Dunkley O, Engel J, Sturm J and Cremers D 2014 Visual-inertial navigation for a camera-equipped 25g nano-quadrotor *IROS2014 Aerial Open Source Robotics Workshop*
- [12] Engel J, Sturm J and Cremers D 2014 Scale-aware navigation of a low-cost quadcopter with a monocular camera *Rob. Auton. Syst.* **62** 1646–56
- [13] Evangelista C, Kraft P, Dacke M, Reinhard J and Srinivasan M V 2010 The moment before touchdown: Landing manoeuvres of the honeybee *Apis mellifera* *J. Exp. Biol.* **213** 262–70
- [14] Expert F, Viollet S and Ruffier F 2011 Outdoor field performances of insect-based visual motion sensors *J. Field Rob.* **28** 529–41
- [15] Expert F and Ruffier F 2015 Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers *Bioinspir. Biomim.* **10** 026003
- [16] Floreano D *et al* 2013 Miniature curved artificial compound eyes *Proc. Natl. Acad. Sci. USA* **110** 9267–72
- [17] Forster C, Pizzoli M and Scaramuzza D 2014 SVO: fast semi-direct monocular visual odometry *IEEE Int. Conf. on Robotics and Automation (ICRA)*
- [18] Franceschini N, Pichon J M, Blanes C and Brady J M 1992 From insect vision to robot vision *Phil. Trans. Biol. Sci.* **337** 283–94
- [19] Fuller S B, Straw A D, Peek M Y, Murray R M and Dickinson M H 2014 Flying drosophila stabilize their vision-based velocity controller by sensing wind with their antennae *Proc. Natl Acad. Sci. USA* **111** E1182–91
- [20] Fuller S B, Helbling E F, Chirarattananon P and Wood R J 2014 Using a mems gyroscope to stabilize the attitude of a fly-sized hovering robot *IMAV 2014: Int. Micro Air Vehicle Conf. and Competition*
- [21] Gibson J J 1979 *The Ecological Approach to Visual Perception* (Boston, MA: Houghton Mifflin)
- [22] Hattenberger G, Bronz M and Gorraz M 2014 Using the paparazzi UAV system for scientific research *IMAV 2014: Int. Micro Air Vehicle Conf. and Competition*
- [23] Helbling E F, Fuller S B and Wood R J 2014 Pitch and yaw control of a robotic insect using an onboard magnetometer *IEEE Int. Conf. on Robotics and Automation (ICRA)* 5516–22 pages
- [24] Hérisse B, Hamel T, Mahony R and Russotto F-X 2012 Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow *IEEE Trans. Robo.* **28** 77–89
- [25] Izzo D and de Croon G C H E 2013 Nonlinear model predictive control applied to vision-based spacecraft landing *EuroGNC 2013 (Delft, Netherlands)*
- [26] Izzo D and de Croon G C H E 2012 Landing with time-to-contact and ventral optic flow estimates *J. Guidance Control Dynamics* **35** 1362–7
- [27] Izzo D, Weiss N and Seidl T 2011 Constant optic-flow lunar landing: optimality and guidance *ALAA J. Guidance Control Dynamics* **34** 1383–95
- [28] Kendoul F 2014 Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems *Int. J. Rob. Res.* **33** 237–67
- [29] Kral K 1998 Side-to-side head movements to obtain motion depth cues: a short review of research on the praying mantis *Behav. Processes* **43** 71–7
- [30] Lee D N, Davies M N O, Green P R and van der Weel F R 1993 Visual control of velocity of approach by pigeons when landing *J. Exp. Biol.* **180** 85–104

- [31] Lucas B D and Kanade T 1981 An iterative image registration technique with an application to stereo vision *Proc. of Imaging Understanding Workshop* p121–130
- [32] Lynen S, Achtelik M W, Weiss S, Chli M and Siegwart R 2013 Robust and Modular multi-sensor fusion approach applied to mav navigation *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*
- [33] Ma K Y, Chirarattananon P, Fuller S B and Wood R J 2013 Controlled flight of a biologically inspired, insect-scale robot *Science* **340** 603–7
- [34] Miao T and Seborg D E 1999 Automatic detection of excessively oscillatory feedback control loops *Int. Conf. on Control Applications (Hawaii, USA)* vol 37, p 38
- [35] Orchard G, Bartolozzi C and Indiveri G 2009 Applying neuromorphic vision sensors to planetary landing tasks *Biomedical Circuits and Systems Conf. BIOCAS 2009* 201–4 IEEE
- [36] Remes B D W, Hensen D, Van Tienen F, De Wagter C, Van der Horst E and de Croon G C H E 2013 Paparazzi: how to make a swarm of parrot ar drones fly autonomously based on GPS *Proc. of the Int. Micro Air Vehicle Conf. and Flight Competition Toulouse, France*
- [37] Ruffier F and Franceschini N 2005 Optic flow regulation: the key to aircraft automatic guidance *Rob. Auton. Syst.* **50** 177–94
- [38] Rzucidło P 2007 The detection of pilot-induced oscillations *Aviation* **11** 15–22
- [39] Chirarattananon P, Fuller S, Helbling E and Wood R 2014 Using a gyroscope to stabilize the attitude of a fly-sized hovering robot *Int. Micro Air Vehicle Competition and Conf. 2014* (Netherlands: Delft) 102–9
- [40] Shi J and Tomasi C 1994 Good features to track *CVPR* pp 593–600
- [41] Song Y M *et al* 2013 Digital cameras with designs inspired by the arthropod eye *Nature* **497** 95–9
- [42] Srinivasan M, Zhang S, Lehrer M and Collett T 1996 Honeybee navigation en route to the goal: visual flight control and odometry *J. Exp. Biol.* **199** 237–44
- [43] Tammero L F and Dickinson M H 2002 Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly *Drosophila melanogaster* *J. Exp. Biol.* **205** 2785–98
- [44] Theys B, Dimitriadis G, Andrianne T, Hendrick P and De Schutter J 2014 Wind tunnel testing of a vtol mav propeller in tilted operating mode *2014 Int. Conf. on Unmanned Aircraft Systems (ICUAS)* pp 1064–72
- [45] Valette F, Ruffier F, Viollet S and Seidl T 2010 Biomimetic optic flow sensing applied to a lunar landing scenario *IEEE International Conference on Robotics and Automation (ICRA 2010)* 2253–60
- [46] De Wagter C, Tijmons S, Remes B D W and de Croon G C H E 2014 Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system *2014 IEEE International Conference on Robotics and Automation (ICRA 2014)* pp 4982–7
- [47] Yildiz Y, Kolmanovsky I V and Acosta D 2011 A control allocation system for automatic detection and compensation of phase shift due to actuator rate limiting *American Control Conf. (ACC), 2011* 444–9 IEEE