

## Distance and velocity estimation using optical flow from a monocular camera

Ho, Hann Woei; de Croon, Guido C.H.E.; Chu, Qiping

**DOI**

[10.1177/1756829317695566](https://doi.org/10.1177/1756829317695566)

**Publication date**

2017

**Document Version**

Final published version

**Published in**

International Journal of Micro Air Vehicles

**Citation (APA)**

Ho, H. W., de Croon, G. C. H. E., & Chu, Q. (2017). Distance and velocity estimation using optical flow from a monocular camera. *International Journal of Micro Air Vehicles*, 9(3), 198-208.  
<https://doi.org/10.1177/1756829317695566>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Distance and velocity estimation using optical flow from a monocular camera

Hann Woei Ho<sup>1,2</sup>, Guido CHE de Croon<sup>1</sup> and Qiping Chu<sup>3</sup>

## Abstract

Monocular vision is increasingly used in micro air vehicles for navigation. In particular, optical flow, inspired by flying insects, is used to perceive vehicle movement with respect to the surroundings or sense changes in the environment. However, optical flow does not directly provide us the distance to an object or velocity, but the ratio of them. Thus, using optical flow in control involves nonlinearity problems which add complexity to the controller. To deal with that, we propose an algorithm that estimates distance and velocity of the vehicle based on optical flow measured from a monocular camera and the knowledge of control inputs. This algorithm applies an extended Kalman filter to state estimation and uses the estimates for landing control. We implement and test our algorithm in computer simulation and on board a Parrot AR.Drone 2.0 to demonstrate its feasibility for micro air vehicles landings. Results of the simulation and multiple flight tests show that the algorithm is able to estimate height and velocity of the micro air vehicles accurately, and achieves smooth landings with these estimates, even in windy outdoor environments.

## Keywords

Distance estimation, monocular vision, optical flow, efference copy, autonomous landing

Date received: 5 December 2016; accepted: 27 January 2017

## Introduction

Micro air vehicles (MAVs) have gained in popularity in recent years. Due to their small size, they are easier and safer to use than large unmanned aerial vehicles. However, the size and weight constraints make the MAVs more challenging to be deployed for autonomous missions. To deal with the problems, it requires reducing the amount of payload or sensors and developing more intelligent and efficient algorithms.

Optical flow, which can be extracted from a monocular camera, provides a very promising solution for miniaturization. Optical flow refers to the apparent visual motion of objects in a scene relative to an observer.<sup>1</sup> This information tells us not only how fast the camera moves, but also how close it is relative to the things it sees. Flying insects use it as the main visual cue for sensing their own movements and surrounding objects while flying. They can perform complex tasks, such as landing, by only relying on this visual input and using limited neural resources. For instance, honeybees heavily use optical flow to perceive the environment and avoid dangerous objects while flying.<sup>2,3</sup>

To accomplish vertical landings, honeybees use a divergent pattern of optical flow or the so-called flow divergence. By keeping the flow divergence constant, honeybees can perform smooth landings. This strategy leads to exponential decay of both height and velocity to zero at touchdown, and it is ideal for MAVs landings.<sup>4–7</sup> Often, a straightforward proportional or proportional integral feedback controller is used for MAVs to perform constant flow divergence landing.

However, in our previous studies, we found that fixed-gain controls used in constant flow divergence can lead to instability of the landing.<sup>8,9</sup> This is because

<sup>1</sup>Micro Air Vehicle laboratory of the Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

<sup>2</sup>Faculty of Aerospace Engineering, Universiti Sains Malaysia, Nibong Tebal, Malaysia

<sup>3</sup>Control and Simulation Section of the Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

## Corresponding author:

Hann Woei Ho, Micro Air Vehicle laboratory of the Faculty of Aerospace Engineering, Delft University of Technology, Delft 2629HS, The Netherlands and the Faculty of Aerospace Engineering, Universiti Sains Malaysia, Nibong Tebal 14300, Malaysia.  
Email: h.w.ho@tudelft.nl



the critical control gain is directly proportional to the height.<sup>8</sup> Most of the current approaches to solving this problem are to use a gain-scheduling technique which is actually designed according to the knowledge of an initial height.<sup>10,11</sup> However, if no additional sensor is installed, the height is unknown. Besides, if the initial height deviates too much from the one which is used to design the gain-scheduling structure, this method will not work properly. Thus, the knowledge of the height is important for optical flow control.

In fact, optical flow does not give us the absolute distance to a surface or velocity of the camera directly. There are four known methods to estimate distances from optical flow. First, one can add sensors that directly or indirectly measure distance, velocity, or accelerations. The right type of filtering and fusion can then give distance estimates. For instance, some studies include an inertial measurement unit (IMU), a camera sensor, and a pressure sensor, and use a Kalman-based sensor fusion technique to estimate distances.<sup>12,13</sup>

Second, when performing a perfect constant flow divergence landing, the control input (the thrust) or so-called efference copy follows a specific monotonously decreasing function over time, which can be used to estimate distances.<sup>14</sup> They demonstrated this approach by moving a camera along a track toward an image scene to first estimate an initial distance. Since the constant flow divergence control will result in an exponential decay of distances, they simplified the estimation problem to an exponential propagation of distances after obtaining the initial value.

Third, a stability approach that detects self-induced oscillation caused by high controller gains and uses these gains to estimate distances was introduced.<sup>8</sup> He showed analytically that there exists a directly proportional relationship between the critical control gain, i.e., the gain which causes instability, and the height. Based on this relationship, the MAV detects self-induced oscillation and uses these gains to estimate the height.

Fourth, by knowing the effect of the control inputs, one can “scale” the optical flow to obtain the distance and velocity. An early study on the dynamic effects in visual servoing showed that the control gain is a function of the distance.<sup>15</sup> It was used in an adaptive control scheme for visual servoing.<sup>16</sup> The relationship also implies that the closed-loop response depends on the distance which opens up the possibility of estimating the distance to a target from closed-loop dynamics.

In this paper, we build upon the principle presented in the fourth method to estimate the height (distance to the ground) and vertical velocity of an MAV. An extended Kalman filter (EKF) is used to estimate the height and vertical velocity of an MAV using the knowledge of the control input in combination with the flow

divergence observed from a monocular camera. This algorithm uses the efference copy to predict the effects of an action and the observed flow divergence to correct the prediction. The significant advantage of this method is that we simplify the nonlinear control of the constant flow divergence to a linear control that uses height and velocity estimates. Thus, this provides the possibility of using some optimization techniques with vision output from a monocular camera to improve the performance of the control.

The remainder of the paper is set up as follows: In the first section, we provide some knowledge about the flow divergence and the constant flow divergence guidance and control strategy. The following section describes the proposed EKF-based height and velocity estimation using the flow divergence and the control inputs, the nonlinear observability analysis of the system, and vision algorithms used to estimate flow divergence with a monocular camera. Then, the next section shows the results of estimation in computer simulation while in the succeeding section we also show the results of multiple landings of an MAV. Finally, a conclusion with future work is drawn.

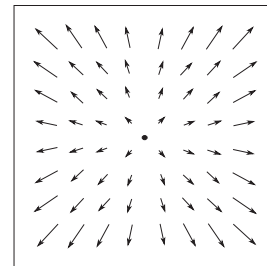
## Background

### Flow divergence

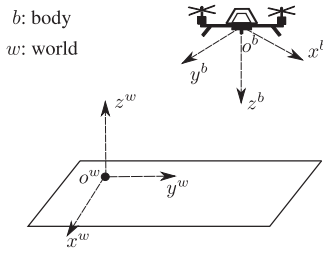
For vertical landing of an MAV, flow divergence ( $D$ ) or visual looming as shown in Figure 1 can be computed as the ratio of its vertical velocity  $V_Z$  to the height from the ground  $Z$

$$D = \frac{V_Z}{Z} \quad (1)$$

When the MAV is approaching the ground, we measure positive  $Z$ , negative  $V_Z$ , and thus  $D < 0$  according to the coordinate systems shown in Figure 2. The camera coordinate system is assumed to coincide with the body coordinate system.



**Figure 1.** Divergence of optical flow vectors (flow divergence) when the observer is approaching a surface.



**Figure 2.** MAV body ( $x^b, y^b, z^b$ ) and world ( $x^w, y^w, z^w$ ) coordinate systems. MAV: micro air vehicle.

### Constant flow divergence guidance and control

Constant flow divergence approach has been used for vertical landing of the MAVs. This approach controls the vertical dynamics of the MAVs by tracking a constant flow divergence. When  $|D|$  equals a positive constant  $c$ , we can derive from equation (1) the height  $Z = Z_0 e^{-ct}$ , and velocity  $V_Z = -cZ_0 e^{-ct}$ , where  $Z_0$  is the initial height. This shows that both height and velocity will decrease exponentially and eventually become zero when touching the ground.

A proportional feedback controller is used to track the desired flow divergence  $D^*$  as shown in equation (2)

$$\mu = K_p(D^* - D) \quad (2)$$

where  $K_p$  is the gain of the proportional controller.

A double integrator system is used to model the dynamics of an MAV toward the ground in one-dimensional space. The continuous state space model can be written as

$$\begin{aligned} \dot{\mathbf{r}}(t) &= f(\mathbf{r}(t), \mu(t)) = \mathbf{A} \cdot \mathbf{r}(t) + \mathbf{B} \cdot \mu(t) \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{r}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mu(t) \end{aligned} \quad (3)$$

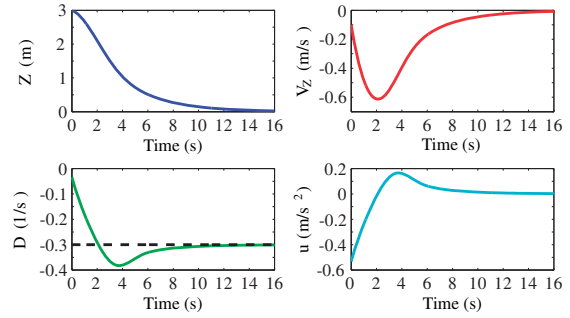
$$y(t) = h(\mathbf{r}(t)) = [r_2(t)/r_1(t)] = D \quad (4)$$

where  $\mathbf{r} = [r_1, r_2]^T = [Z, V_Z]^T$  and  $\mu$  is the control input.

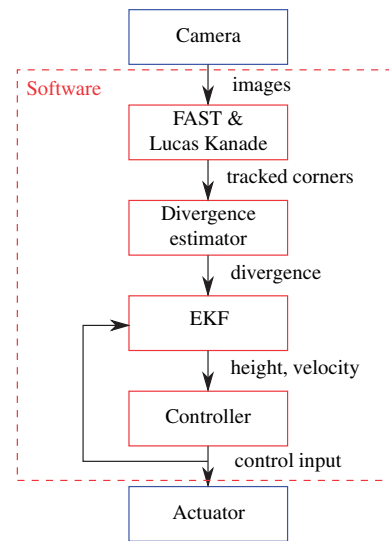
It is clear that the model dynamics in equation (3) are linear but the observation in equation (4) is nonlinear. Figure 3 shows a time response of the system when tracking a constant flow divergence. In this figure, we can see that the MAV accelerates in the first 2.5 s and then decelerates to zero velocity to touch the ground. Both height and velocity of the MAV decrease exponentially to zero in the end.

### EKF-based height and velocity estimation using flow divergence and control input

An overview of the methodology is presented in Figure 4. Images captured by a camera are served as



**Figure 3.** Height  $Z$ , velocity  $V_Z$ , and flow divergence  $D$  of an MAV during landing using constant flow divergence strategy ( $D^* = -0.3 \text{ s}^{-1}$ ) with control input  $\mu$ . MAV: micro air vehicle.



**Figure 4.** An overview of the methodology.

the inputs to the software while the control input commands the actuator to control the MAV. In this section, we describe (1) an EKF algorithm using the flow divergence and efference copies to estimate the height and velocity of an MAV during landing, (2) a nonlinear observability analysis of the system, and (3) a vision method to compute the flow divergence.

### EKF

In practice, the flow divergence is computed using an on-board processor. Therefore, we used a discrete-time EKF to estimate the height and vertical velocity of the MAV. The system model and observation model are shown in equations (5) and (6), respectively

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mu(t)) + \mathbf{w}(t) \quad (5)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (6)$$

where  $f(\cdot)$  and  $h(\cdot)$  are the system matrix and the observation matrix which are obtained from equations (3) and (4), respectively.  $\mathbf{w}(t)$  and  $\mathbf{v}_k$  are the system noise and observation noise which are both assumed to be zero mean multivariate Gaussian process with covariance  $\mathbf{Q}$  and  $\mathbf{R}$ , respectively.

Several computational steps taken in EKF when the update of flow divergence is obtained are shown below:

(a) One-step ahead prediction

$$\hat{\mathbf{x}}_{k|k-1} = \hat{\mathbf{x}}_{k-1|k-1} + \int_{t_{k-1}}^{t_k} f(\mathbf{x}(\tau), \mu(\tau)) d\tau \quad (7)$$

(b) Covariance matrix of the state prediction error vector

$$\mathbf{P}_{k|k-1} = \mathbf{\Phi}_k \mathbf{P}_{k-1|k-1} \mathbf{\Phi}_k^T + \mathbf{\Gamma}_k \mathbf{Q} \mathbf{\Gamma}_k^T \quad (8)$$

where  $\mathbf{\Phi}$  and  $\mathbf{\Gamma}$  are the discretized matrices of  $\mathbf{A}$  and  $\mathbf{B}$  and can be computed as below (their derivations can be found in Appendix 1)

$$\mathbf{\Phi}_k = \begin{bmatrix} 1 & t_k - t_{k-1} \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma}_k = \begin{bmatrix} (t_k - t_{k-1})^2/2 \\ t_k - t_{k-1} \end{bmatrix} \quad (9)$$

(c) Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R})^{-1} \quad (10)$$

where

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} = \begin{bmatrix} -\frac{\hat{x}_{2k|k-1}}{\hat{x}_{1k|k-1}^2} & \frac{1}{\hat{x}_{1k|k-1}} \end{bmatrix}^T \quad (11)$$

(d) Measurement update

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \quad (12)$$

where  $\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$  is called the innovation of EKF.

(e) Covariance matrix of state estimation error vector

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \quad (13)$$

## Nonlinear observability of the estimation

In this subsection, we show that with the flow divergence and the efference copy/control input the system is observable, i.e., any distinct states are distinguishable by applying a bounded measurable input (e.g., a piecewise constant input). To check the observability of the nonlinear system, the observability rank condition can be used.<sup>17</sup>

We first construct the observability algebra which consists of repeated Lie derivatives of the observation model in equation (4),  $h(\mathbf{r}(t)) = r_2(t)/r_1(t)$ , with respect to the model dynamics in equation (3),  $f(\mathbf{r}(t), \mu(t)) = [r_2(t), \mu]^T$ . The system is observable if and only if the Jacobian of the observability algebra is full rank. In our case, the observability algebra  $\mathcal{O}$  can be written as:

$$\begin{aligned} \mathcal{O} &= \begin{bmatrix} L_f^0 h(\mathbf{r}(t)) \\ L_f^1 h(\mathbf{r}(t)) \end{bmatrix} = \begin{bmatrix} h(\mathbf{r}(t)) \\ \frac{\partial h(\mathbf{r}(t))}{\partial \mathbf{r}(t)} \cdot f(\mathbf{r}(t), \mu(t)) \end{bmatrix} \\ &= \begin{bmatrix} r_2(t)/r_1(t) \\ -(r_2(t))^2/(r_1(t))^2 + \mu/(r_1(t)) \end{bmatrix} \end{aligned} \quad (14)$$

Next, we analyze  $\mathcal{O}$  to see whether its Jacobian is full rank. The first function of the observability algebra is  $y(t) = h(\mathbf{r}(t)) = [r_2(t)/r_1(t)] = D$  from equation (4) while its second function is the first Lie derivative with respect to the dynamics which equals to  $-(r_2(t))^2/(r_1(t))^2 + \mu/(r_1(t)) = -(h(\mathbf{r}(t)))^2 + \mu/(r_1(t))$ . From this result, we can clearly see that the system is (locally) observable (i.e., a full rank Jacobian of  $\mathcal{O}$ ) if and only if the control input,  $\mu \neq 0$ , and  $r_1(t) \neq 0$  and  $r_2(t) \neq 0$ . In other words, we are able to estimate the distinct states, i.e.,  $Z$  and  $V_Z$ , by using both flow divergence and control input in the algorithm.

## Features-based flow divergence estimation

In computer simulation, we used equation (1) to compute flow divergence, while in flight tests, we estimated flow divergence based on equation (15) (Proof can be found in Ho et al.<sup>9</sup>). For each image captured by an on-board camera, corners were detected using the FAST algorithm<sup>18,19</sup> and tracked in the next image using the Lucas-Kanade tracker.<sup>20</sup> Then, the image distances between every two corners at one image,  $d_{(t-\Delta t),i}$  and at the next image,  $d_{t,i}$  were computed. By further computing the ratio of  $d_{(t-\Delta t),i} - d_{t,i}$  to  $d_{(t-\Delta t),i}$ , we can measure the expansion and contraction of the flow. We took the average of these ratios, and with a known time interval between these two images  $\Delta t$ , we estimated divergence using equation (15)

$$\hat{D} = \frac{1}{n} \cdot \frac{1}{\Delta t} \sum_{i=1}^n \left[ \frac{d_{(t-\Delta t),i} - d_{t,i}}{d_{(t-\Delta t),i}} \right] \quad (15)$$

where  $n$  is the total number of tracked corners. In practice, the vision output is often noisy. Therefore, we used a low pass filter to reduce the noise in the estimation.

## Computer simulation

Before performing flight tests, we simulated the proposed algorithm presented in the previous section in MATLAB to show the feasibility of the algorithm.

### Landing simulation with simulated control inputs

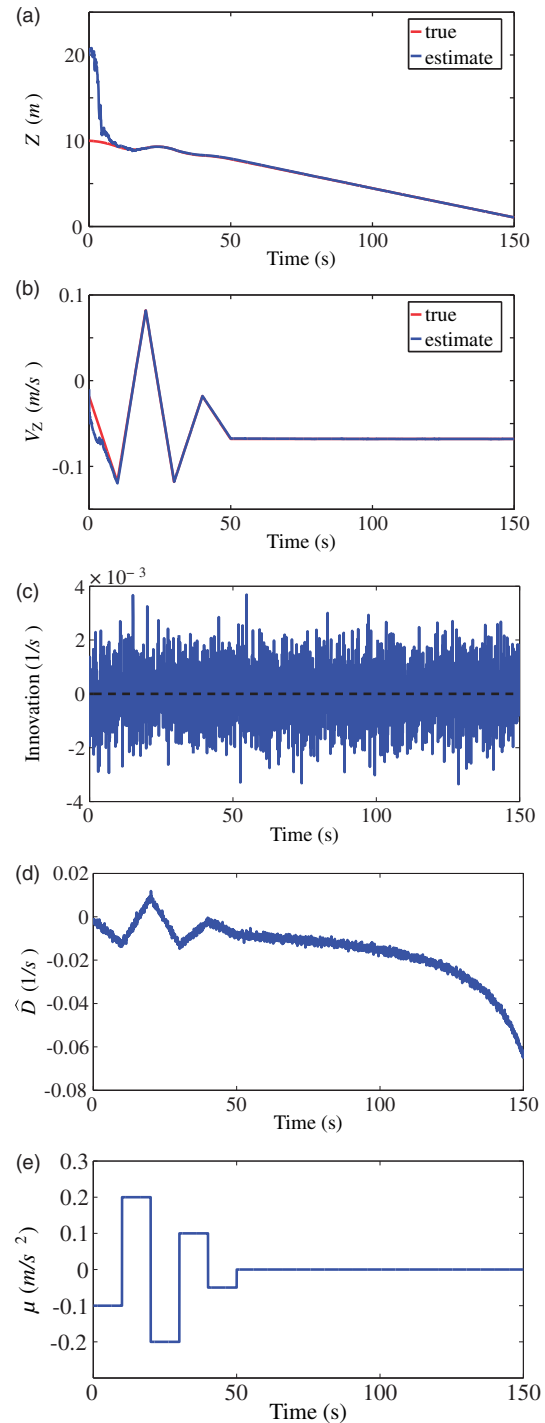
In the simulation, we generated the height and velocity with a timestamp of 0.05 s using a set of control inputs,  $\mu$  as shown in Figure 5(e). These data served as ground truth for validation. The flow divergence measurement was generated using equation (1) with a measurement noise standard deviation of  $0.001 \text{ s}^{-1}$  as illustrated in Figure 5(d).

By using the control input,  $\mu$  and the flow divergence measurement,  $D$ , we estimated the height,  $Z$  and the velocity,  $V_Z$  with the proposed EKF algorithm. Figure 5 shows estimated states ( $Z$  and  $V_Z$ ) and their ground truth, innovation of the EKF, flow divergence measurement, and the control inputs. In this simulation, we can observe that the estimated height and velocity converge and follow the ground truth after a few seconds, even with different initial conditions ( $Z_0$  and  $V_{Z0}$ ) than the actual values. In addition, the innovation of EKF has zero mean implying that the filter is working correctly. Simulation results show that the proposed algorithm is able to estimate the distance to the ground and velocity accurately.

## Experiments results and discussion

We implemented the EKF algorithm and vision algorithms in Paparazzi Autopilot, an open source autopilot software.<sup>21</sup> A Parrot AR.Drone 2.0 equipped with a downward-looking camera was used as a testing platform, and all algorithms were running on board the MAV. We used an *OptiTrack* system to track the position of the MAV in order to provide the ground truth of its height and velocity *only for validation purposes*, and these measurements were *not* used in the estimation. In this section, we show the feasibility of the algorithm by performing three different control strategies for MAVs landings.

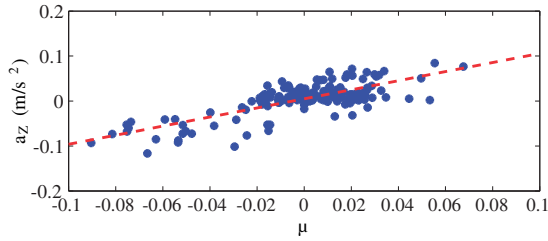
Before executing the algorithm in real-world experiments, we need to check the relationship between the control input  $\mu$  (which is assumed to be the acceleration in the model) and the resulting acceleration  $a_Z$  imposed on our platform, e.g.,  $a_Z = f(\mu)$ . Figure 6 shows the vertical acceleration  $a_Z$  resulted from the control input  $\mu$  used in performing a constant flow divergence



**Figure 5.** EKF-based height and vertical velocity estimation from flow divergence for landing control. (a) Height, (b) velocity, (c) innovation, (d) flow divergence, and (e) control input. EKF: extended Kalman filter.

landing. The acceleration measurement was obtained from the on-board IMU but was *not* used in the following flight tests. Knowing that the variation of the control input  $\mu$  from the hovering command  $\mu_g$  is small (total command to motors  $\mu_T = \mu_g + \mu$ ), a linear





**Figure 6.** Mapping of the control input  $\mu$  to the vertical acceleration  $a_z$  using a linear function ( $a_z = 9.906\mu + 0.047$ ).

function is used to map the control input to the acceleration. From the mapping, we obtained  $a_z = 9.906\mu + 0.047$  plotted as a dashed line in the figure.

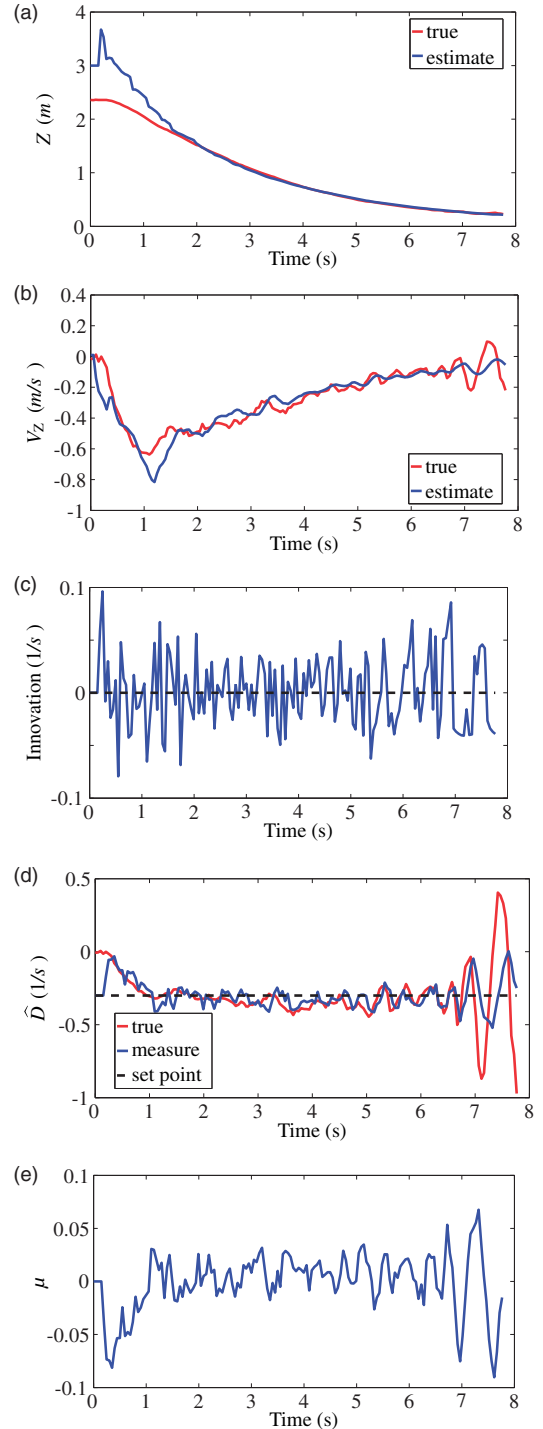
### Landing using constant flow divergence with fixed-gain control

The first strategy we used in the flight tests is the constant flow divergence. A flow divergence set point was tracked for the entire landing. During the landing, the EKF algorithm was running in real time to estimate the height and velocity of the MAV.

Figure 7 shows the estimation results of the landing with a constant divergence of  $-0.3\text{ s}^{-1}$  using a fixed-gain control. The initial value of the height for EKF was set to be  $\hat{Z}_0 = 3\text{ m}$  which was different from the true initial height, i.e.,  $Z_0 \approx 2\text{ m}$ . In the figure, we can see that the height estimate converges to the true height, even though the initial value is different. In addition, the velocity estimate also matches well with the true velocity.

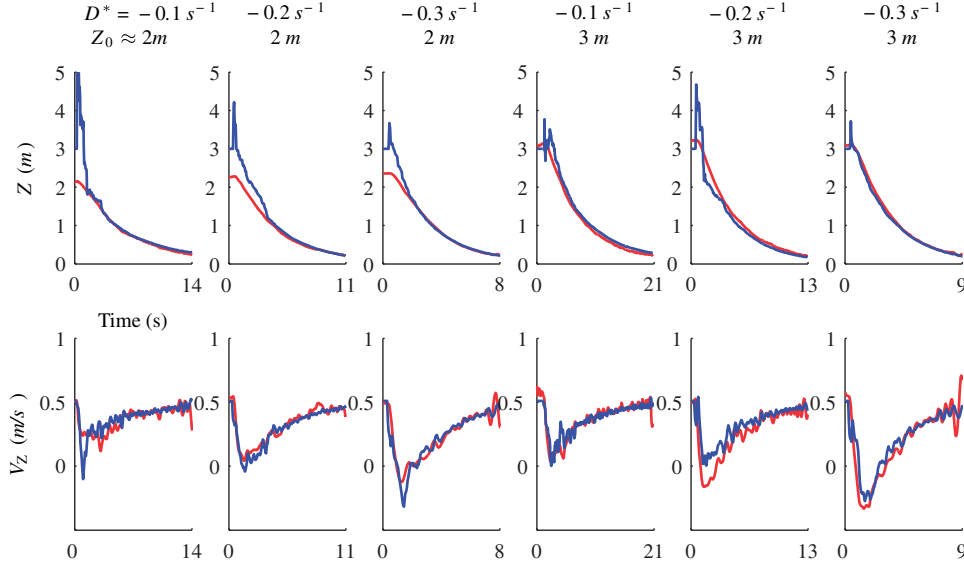
From the landing results, we can observe that this strategy exponentially decreases both height and velocity to nearly zero by only tracking a constant flow divergence. In fact, the drawback of vision algorithms is that when the image is too dark, the algorithms will not work properly. For instance, when the camera is too close to the ground, the flow divergence estimate becomes incorrect. To deal with that, we constantly decrease the trim throttle when the MAV is very close to the ground ( $Z < 0.2\text{ m}$ ) to allow the MAV for touch-down. Another fact is that a fine-tuned control gain or gains adapted with the height is needed to solve the instability issue.<sup>9</sup> This issue can be seen from the measured flow divergence in Figure 7(d) when the MAV is close to the ground.

To show the reliability of the proposed algorithm, we performed multiple landings with different flow divergence set points ( $D^* = -0.1, -0.2, -0.3\text{ s}^{-1}$ ) at different initial heights ( $Z_0 \approx 2, 3\text{ m}$ ). The same initial guess of the height (3 m) was used in the EKF algorithm for these flight tests. Figure 8 shows the height and velocity estimates using EKF algorithm in different landings. In this figure, we can see that both height and



**Figure 7.** Landing with constant flow divergence ( $D^* = -0.3\text{ s}^{-1}$ ) with EKF-based height and velocity estimation. (a) Height, (b) velocity, (c) innovation, (d) flow divergence, and (e) control input. EKF: extended Kalman filter.

velocity estimates match well with the true values. With a larger set point, the MAV landed faster with the exponential decrease in both height and velocity. Besides, these results also show that a good guess of the initial height has a faster convergence of the estimates.



**Figure 8.** Multiple landings with different flow divergence set points ( $D^* = -0.1, -0.2, -0.3 \text{ s}^{-1}$ ) at different initial heights ( $Z_0 = 2, 3 \text{ m}$ ). The estimates are shown with blue lines while their true values are represented by red lines.

The height and velocity estimated from EKF can then be adapted to tune the controller gain in real time for the constant divergence landing. This will be discussed in the coming section. By doing that, oscillations due to improperly chosen gain can be reduced, and thus a smooth landing can be achieved by only using a monocular camera.

### Landing using constant flow divergence with height-adapted control

Experiments presented in the previous subsection were performed using a manually tuned control gain in order to achieve stable landings. In fact, the critical control gain, i.e., the control gain which causes instability, is directly proportional to the height.<sup>8</sup> This means that if a fixed gain is used in constant flow divergence landing, oscillation will occur at a certain height. When the height is known from the proposed algorithm, it can be used to automatically adjust the control gain in such a way that the oscillation is eliminated.<sup>9</sup> This second strategy is used to keep the advantages of using constant flow divergence landing, i.e., achieving an exponential landing profile by just tracking a desired flow divergence set point and avoid the instability due to a fixed-gain control.

By letting  $K_p = f(Z) = k \cdot Z$  where  $k$  is a positive constant, the proportional controller in equation (2),  $\mu = K_p(D^* - D)$  becomes

$$\mu = k \cdot Z \left( D^* - \frac{V_Z}{Z} \right) = kD^*Z - kV_Z = -\mathbf{k} \cdot \mathbf{r}(t) \quad (16)$$

where  $\mathbf{k} = [k_1 \quad k_2] = [-kD^* \quad k]$ .

This controller simplifies the problem to a linear control problem. To find  $k$ , we can first solve this linear control problem using the pole-zero analysis. From equation (3), the model dynamics becomes

$$\begin{aligned} \dot{\mathbf{r}}(t) &= \mathbf{A} \cdot \mathbf{r}(t) + \mathbf{B} \cdot (-\mathbf{k} \cdot \mathbf{r}(t)) = (\mathbf{A} - \mathbf{B} \cdot \mathbf{k}) \cdot \mathbf{r}(t) \\ &= \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} \cdot \mathbf{r}(t) \end{aligned} \quad (17)$$

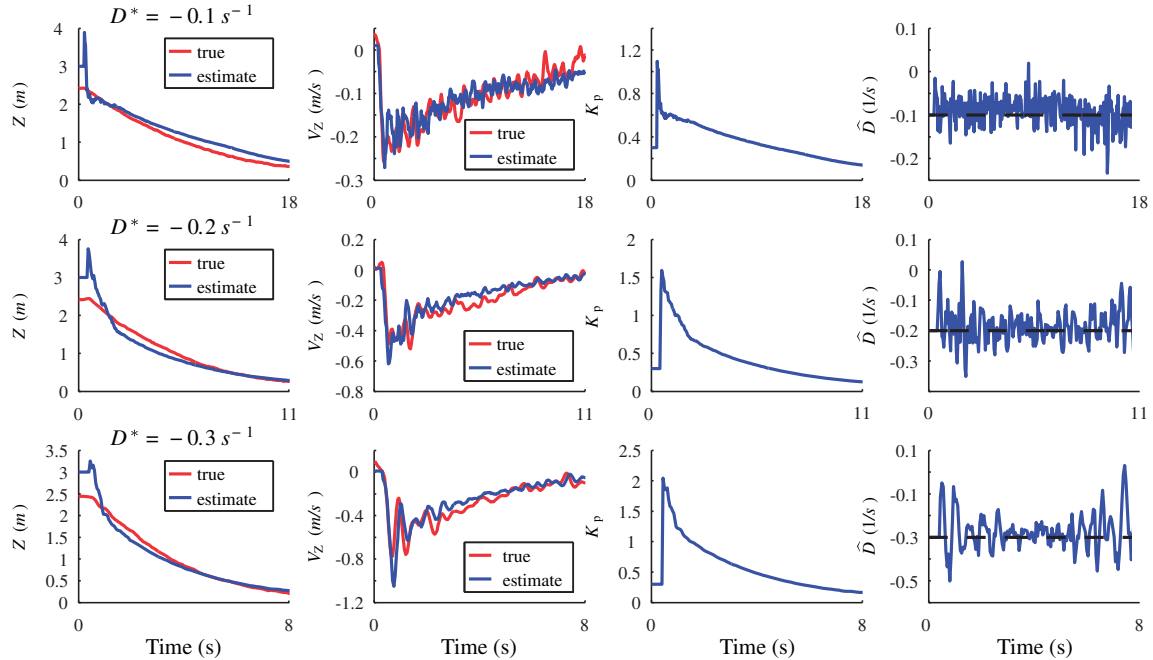
To achieve stable conditions, all the poles should lie in the left half of the s-plane. The poles of the system are obtained as shown below

$$\lambda_{1,2} = \frac{-k_2 \pm \sqrt{k_2^2 - 4k_1}}{2} \quad (18)$$

Therefore, for a stable system,  $k_2^2 \geq 4k_1$ . By substituting  $k_1 = -kD^*$  and  $k_2 = k$ , we get  $k^2 \geq -4kD^*$  and thus  $k(k - 4c) \geq 0$ , where  $c = -D^*$ . Since  $k > 0$ ,  $k \geq 4c$ .

Three flight tests were performed at different flow divergence set points ( $D^* = -0.1, -0.2, -0.3 \text{ s}^{-1}$ ) shown in Figure 9. The initial guess of the height for EKF was set to be different from the true height in order to test the convergence of the estimates. When the landing was activated, the control gain was adapted to the height according to  $K_p = k \cdot \hat{Z}$ . Note that the value of  $K_p$  used in the real flight needs to be mapped according to the result shown in Figure 6. Since the initial height was selected to be much higher than the true height, it was expected that a slight oscillation occurred at the beginning of the landing. After





**Figure 9.** Height-adapted control for constant flow divergence landings with different flow divergence set points ( $D^* = -0.1, -0.2, -0.3 \text{ s}^{-1}$ ).

the height estimate converged to the actual height, the landing was smooth as the correct gain was used. In addition, it can be seen from the last column of Figure 9 that the tracking of these three different  $D^*$  is good. This strategy ensures a stable and high-performance landing.

### Landing using height and velocity from EKF

The third strategy we used for landing is tracking a height profile using the estimates from EKF. To this end, the landing begins by (1) giving an initial excitation to the MAV, or (2) using constant divergence strategy, for the first few seconds. This can ensure that the EKF converges and the controller tracks the right height estimate.

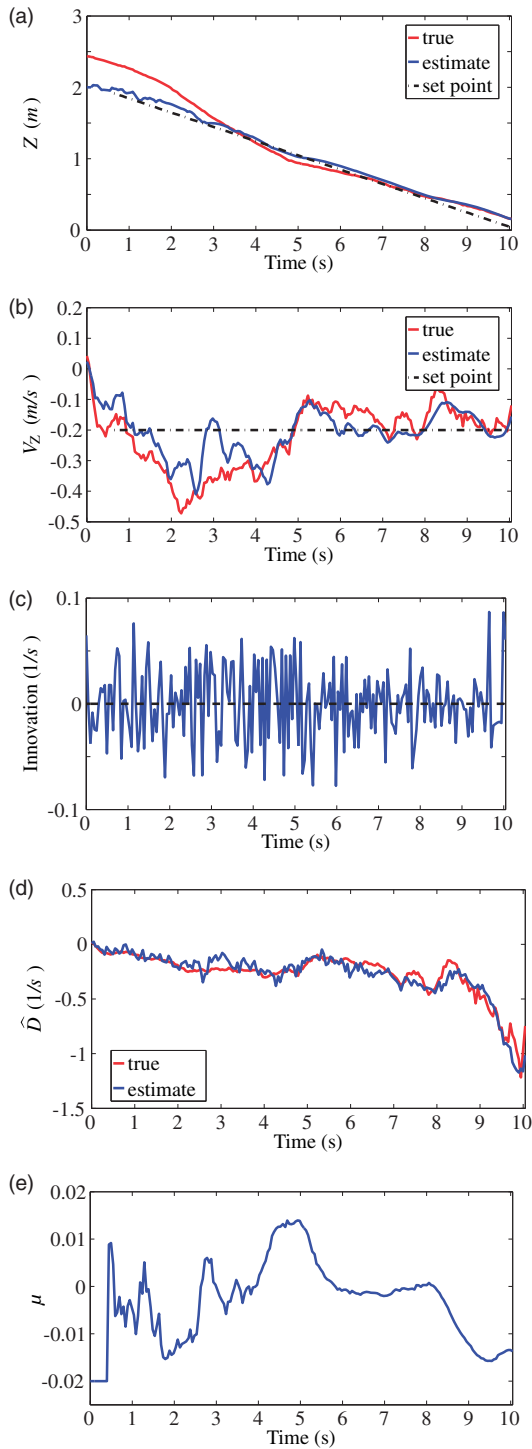
This first initialization was performed in an indoor flight test by giving a block input as the excitation as shown in Figure 10(e) to the MAV for the first 0.5 s. This allowed the MAV to go down or move in order to “observe” flow divergence and initialize the EKF algorithm. After that, the controller used the estimates from the EKF to track a desired profile or set points for landing.

Figure 10 shows the height and velocity estimates from EKF compared with their ground truth, the innovation of EKF, the flow divergence from the vision algorithms and the ground truth, and the control input. After giving the initial excitation, the MAV started to track a height profile represented by the

black dash-dot line. This profile was generated using  $x_{1,k}^* = x_{1,k-1}^* + x_2^* \cdot \Delta t$  with  $x_{1,0}^* = \hat{Z}_0$  and  $x_2^* = -0.2 \text{ m s}^{-1}$ . The results show that the height and velocity can be estimated accurately using the proposed EKF algorithm and further used in the controller for landing. The zero mean innovation of the EKF also tells us that the filter was working properly.

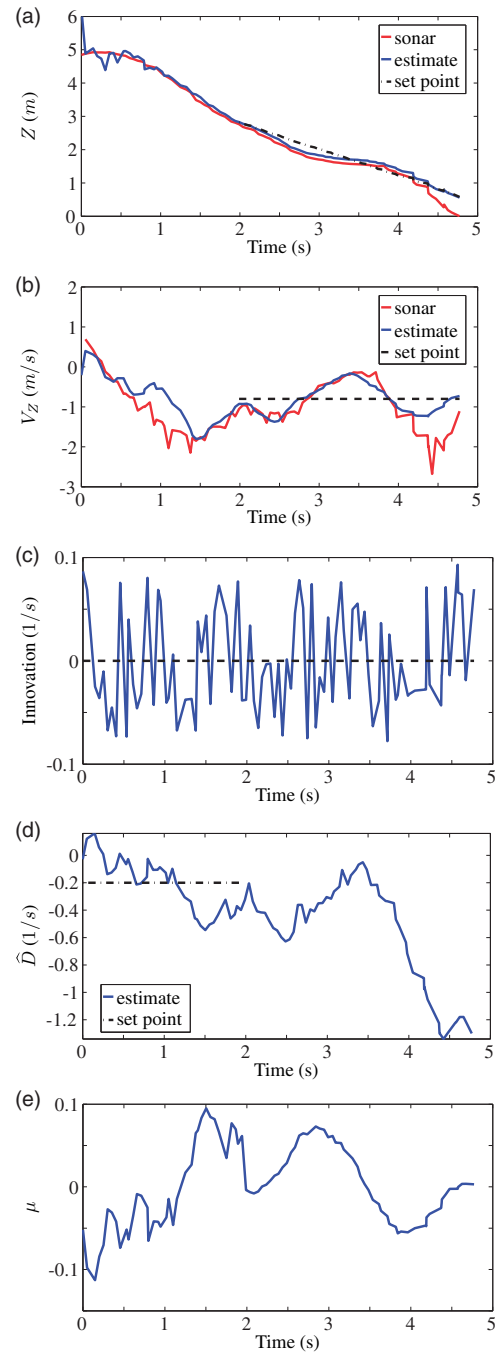
The second initialization was conducted in an outdoor flight by tracking a desired divergence for the first few seconds as shown in Figure 11. Since the *OptiTrack* system is not available for outdoor flight, we can only use the onboard sensor, such as sonar, to provide the true height and velocity ( $dZ/dt$ ) for accuracy comparison.

The landing started at 5 m, and the desired divergence of  $-0.2 \text{ s}^{-1}$  was tracked. After 2 s, the controller was switched to track a height profile as presented in Figure 11(a) which was generated based on the velocity estimate at the instance when the controller was switched ( $\approx -0.8 \text{ m s}^{-1}$ ). In fact, the height profile can also be created based on the desired velocity (see Figure 10). The results show that both height and velocity estimates are accurately estimated compared to the measurements from sonar. In addition, the height estimate can also be used in landing control. It can also be observed from Figure 11 that the estimates converge soon after the landing starts. Thus, the time interval for initialization can also be shortened. Note that the previous indoor flights were performed to validate the proposed algorithm using the highly accurate



**Figure 10.** Landing with a linear controller using estimates from EKF (indoor flight). (a) Height, (b) velocity, (c) innovation, (d) flow divergence, and (e) control input. EKF: extended Kalman filter.

*OptiTrack* system while the outdoor flight was conducted to show the robustness of the algorithm to a windy outdoor environment. Some videos of the flight tests are available online.<sup>a</sup>



**Figure 11.** Landing with a linear controller using estimates from EKF (outdoor flight). (a) Height, (b) velocity, (c) innovation, (d) flow divergence, and (e) control input. EKF: extended Kalman filter.

In this paper, we presented three landing strategies using (1) constant flow divergence with fixed-gain control, (2) constant flow divergence with height-adapted control, and (3) height and velocity estimates. All these strategies can provide a smooth landing solution for MAVs. The first strategy, however, requires a perfectly tuned control gain to avoid instability. The second

strategy solves this manual tuning problem by adapting height estimate from EKF to the control gain. This allows stable landings without affecting the tracking performance. Due to fast convergence of height, we only observe a slight oscillation at the start of the landings. Lastly, the third strategy uses the height and velocity from EKF directly for landing. Although it requires an initial excitation for observing optical flow, this strategy lands the MAV by following a desired landing profile and allows further performance improvement using gain optimization techniques.

### Discussion on wind effect and rejection

In principle, external disturbance, such as the wind, has no problem in control as it can be rejected easily. For instance, when using a proportional feedback control with height estimates, the steady-state error due to wind can occur, but it can be eliminated using the analysis of final value theorem on our model with the wind. From this analysis, we found that in order to reject for instance a constant wind, the proportional control gain should be  $K_P = d_0/x(\infty)$ , where  $x(\infty)$  is the required steady-state error, and  $d_0$  is the wind disturbance estimate.

However, for state estimation, the wind as the external force is not accounted for in our model at the moment. The practical experiment performed in a windy outdoor environment shows that this factor might not be problematic. However, a more thorough investigation and a solution for this problem might be necessary. For instance, we can augment the state vector in equation (3) with the wind disturbance as an additional state, i.e.,  $\dot{d}_{wind} = w$  ( $w$  can be modeled by the random walk<sup>22,23</sup>) and modify the second state to be  $\dot{V}_Z = \mu + d_{wind}$ . By doing that, the wind factor is included in the model, and thus the estimation can be improved in the presence of the wind.<sup>24</sup> Both discussions to deal with external disturbance will be the future work of this study.

### Conclusion

In conclusion, we proposed an algorithm to use an EKF to estimate the height and vertical velocity of an MAV from the flow divergence and the knowledge of the control input while approaching a surface and use these estimates for landing control. This algorithm was tested in computer simulation as well as in multiple flight tests in indoor and windy outdoor environments. The results show that the proposed EKF approach managed to estimate the height and velocity of the MAV accurately compared to the ground truth provided by the external cameras. In addition, the estimates were used in the controller to smoothly land

the MAV. The proposed approach avoids the complexity of having nonlinearity in the constant flow divergence-based landing by “splitting” the flow divergence into the height and velocity which allows the use of a linear controller. For future work, an augmented state Kalman filter can be used to improve the estimation when strong external disturbances are involved. Also, the algorithm opens up the possibilities to use gain optimization techniques to improve the control performance.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

### Note

a. Experiment videos: <https://goo.gl/KiJurr>.

### References

1. Gibson JJ (1950) *The perception of the visual world*. Boston: Houghton Mifflin.
2. Srinivasan M, Zhang S, Lehrer M, et al. Honeybee navigation en route to the goal: visual flight control and odometry. *J Exp Biol* 1996; 199: 237–244.
3. Tammero LF and Dickinson MH. The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *J Exp Biol* 2002; 205: 327–343.
4. McCarthy C, Barnes N and Mahony R. A robust docking strategy for a mobile robot using flow field divergence. *IEEE Trans Robot* 2008; 24: 832–842.
5. Herisse B, Russotto FX, Hamel T, et al. Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. In: *IEEE/RSJ international conference on intelligent robots and systems, 2008. IROS 2008*, Nice, France, 2008, pp.801–806. IEEE.
6. Ruffier F and Franceschini N. Aerial robot piloted in steep relief by optic flow sensors. In: *IEEE/RSJ international conference on intelligent robots and systems, 2008. IROS 2008*. IEEE, 2008, pp.1266–1273..
7. Alkowitz MT, Becerra VM and Holderbaum W. Bioinspired autonomous visual vertical control of a quadrotor unmanned aerial vehicle. *J Guid Control Dyn* 2014; 38: 249–262.
8. de Croon GCHE. Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy. *Bioinspir Biomimet* 2016; 11: 016004.
9. Ho HW, de Croon GCHE, van Kampen EJ, et al. Adaptive control strategy for constant optical flow divergence landing. ArXiv preprint arXiv:1609.06767, 2016.
10. Kendoul F. Four-dimensional guidance and control of movement using time-to-contact: application to

- automated docking and landing of unmanned rotorcraft systems. *Int J Robot Res* 2014; 33: 237–267.
11. de Croon GCHE, Alazard D and Izzo D. Controlling spacecraft landings with constantly and exponentially decreasing time-to-contact. *IEEE Trans Aerosp Electr Syst* 2015; 51: 1241–1252.
  12. Lynen S, Achtelik MW, Weiss S, et al. A robust and modular multi-sensor fusion approach applied to MAV navigation. In: *2013 IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 2013, pp.3923–3929. IEEE.
  13. Engel J, Sturm J and Cremers D. Scale-aware navigation of a low-cost quadcopter with a monocular camera. *Robot Auton Syst* 2014; 62: 1646–1656.
  14. van Breugel F, Morgansen K and Dickinson MH. Monocular distance estimation from optic flow during active landing maneuvers. *Bioinspir Biomimet* 2014; 9: 025002.
  15. Corke PI and Good MC. Dynamic effects in high-performance visual servoing. In: *1992 IEEE international conference on robotics and automation, 1992. Proceedings*. Nice, France, 1992, pp.1838–1843. IEEE.
  16. Asl HJ, Oriolo G and Bolandi H. An adaptive scheme for image-based visual servoing of an underactuated UAV. *Int J Robot Autom* 2014; 29: 92–104.
  17. Hermann R and Krener AJ. Nonlinear controllability and observability. *IEEE Trans Autom Control* 1977; 22: 728–740.
  18. Rosten E and Drummond T. Fusing points and lines for high performance tracking. In: *Tenth IEEE international conference on computer vision, 2005. ICCV 2005*, Vol. 2. Beijing, China, 2005, pp.1508–1515. IEEE.
  19. Rosten E and Drummond T. Machine learning for high-speed corner detection. In: *Computer Vision–ECCV 2006*, Graz, Austria, 2006, pp.430–443. Springer.
  20. Bouquet JY (2000) *Pyramidal implementation of the Lucas Kanade feature tracker*. Intel Corporation, Microsoft Research Labs, Tech. Rep.
  21. Remes BDW, Hensen D, van Tienen F, et al. Paparazzi: how to make a swarm of Parrot AR Drones fly autonomously based on GPS. In: *International Micro Air Vehicle Conference and Flight Competition (IMAV2013)*, Toulouse, France, 2013, pp.17–20..
  22. Hsieh CS. Robust two-stage Kalman filters for systems with unknown inputs. *IEEE Trans Autom Control* 2000; 45: 2374–2378.
  23. Park SH, Kim PS, Kwon OK, et al. Estimation and detection of unknown inputs using optimal FIR filter. *Automatica* 2000; 36: 1481–1488.
  24. Lu P, van Kampen EJ, de Visser C, et al. Nonlinear aircraft sensor fault reconstruction in the presence of disturbances validated by real flight data. *Control Eng Pract* 2016; 49: 112–128.
  25. DeCarlo RA. *Linear systems: a state variable approach with numerical implementation*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc, 1989..

## Appendix I: Discretization of linear state space models

The continuous state space model is given as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (19)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (20)$$

The discretized state space model can be expressed below

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{u}[k] \quad (21)$$

$$\mathbf{y}[k] = \mathbf{C}_d\mathbf{x}[k] + \mathbf{D}_d\mathbf{u}[k] \quad (22)$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}dt} \quad (23)$$

$$\mathbf{\Gamma} = \left( \int_{\tau=0}^{dt} e^{\mathbf{A}\tau} d\tau \right) \mathbf{B} \quad (24)$$

$$\mathbf{C}_d = \mathbf{C} \quad (25)$$

$$\mathbf{D}_d = \mathbf{D} \quad (26)$$

$dt$  is the sampling time. We can find both  $\mathbf{\Phi}$  and  $\mathbf{\Gamma}$  at the same time by computing the matrix exponential<sup>25</sup>

$$\exp\left(\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} dt\right) = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (27)$$