Distance and velocity estimation using optical flow from a monocular camera

Ho, Hann Woei; de Croon, Guido; Chu, Qiping

Publication date
2016

Document Version
Accepted author manuscript

Published in
Proceedings of the International Micro Air Vehicles Conference and Competition 2016

Distance and velocity estimation using optical flow from a monocular camera

H.W. Ho*, G.C.H.E. de Croon, and Q.P. Chu
Delft University of Technology, The Netherlands

ABSTRACT

Monocular vision is increasingly used in Micro Air Vehicles for navigation. In particular, optical flow, inspired by flying insects, is used to perceive vehicles' 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 to use an extended Kalman filter to estimate the distance and velocity of the vehicles from optical flow while approaching a surface, and then use these estimates for control. We implement and test our algorithm in computer simulation and on-board a Parrot AR.Drone 2.0 to demonstrate its feasibility for MAVs landings. Both results show that the algorithm is able to estimate height and velocity of the MAV accurately.

1 INTRODUCTION

Micro air vehicles (MAVs) have been widely used due to its small size and enormous capabilities. These two advantages are also two of the biggest challenges to the MAVs as the amount of payloads can be carried on-board and the computing power are limited [1]. Therefore, light-weight sensors which can provide rich information about the surrounding of the MAVs are highly desired. In addition, effective and efficient algorithms are needed to estimate useful inputs to the MAVs from the sensory information [2].

In fact, flying insects have the capabilities to perform complex tasks by only using their bare eyes and a tiny brain. For instance, honeybees heavily use optical flow to perceive the environment and avoid dangerous objects while flying [3, 4]. Optical flow refers to visual relative motion of the observer's eyes or a camera with respect to the objects in the scene [5]. This information tells us not only how fast the camera moves, but also how close it is relatives to the things it sees. However, optical flow does not give us the absolute distance to a surface or velocity of the camera directly.

For a particular case when an MAV is approaching the ground during landing, a downward-facing camera on-board the MAV perceives divergent pattern of the optical flow, or so-called flow divergence. Flow divergence is the ratio of velocity to distance to a surface in the same direction and it is the reciprocal to time-to-contact. By only keeping the flow divergence constant, the MAV is able to perform smooth landing using a monocular camera [6, 7, 8, 9]. In fact, using directly the flow divergence in control involves the problem of nonlinearity in the controller which is complex, and is sensitive when the MAV is close to the ground [10, 11].

The common state estimation techniques using a visual-inertial system in MAVs combined the advantages of using an Inertial Measurement Unit (IMU) and a camera sensor for real-time computation [12, 13, 14]. Although these sensors are usually installed on-board the MAVs, it is definitely an added advantage for MAVs if only a monocular camera is used for navigation. Some recent approaches used 'efference copies' or available control input information fused with flow divergence, to estimate distance in a constant flow divergence control [15, 16]. In [15], they used the flow divergence and the control input that moved a camera along a track toward an image scene to first estimate the initial distance. Since the constant flow divergence control will result in exponential decay of distance, they simplified the estimation problem to exponential propagation of distance after obtaining the initial value. In [16], a stability approach that detects self-induced oscillation caused by high controller gains and uses these gains to estimate distance was introduced.

The main contribution of this paper is to estimate height (distance to ground) and vertical velocity of an MAV using a monocular camera while approaching a surface, e.g. in the landing phase. The difference from the above-mentioned approaches is that we propose a straightforward algorithm that uses an extended Kalman filter (EKF) to 'split' the height and vertical velocity of an MAV from flow divergence. 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 Section 2, we provide some knowledge about flow divergence and constant flow divergence guidance and control strategy. Section 3 describes the proposed EKF-based height and velocity estimation using flow divergence and vision algorithms used to estimate flow divergence with a monocular camera.

*Email address(es): h.w.ho@tudelft.nl

Then, Section 4 shows the results of estimation in computer simulation while in Section 5 we also show the estimation results and landing using the estimates in flight. Finally, a conclusion with future works is drawn in Section 6.

2 Background

2.1 Flow Divergence

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

$$D = \frac{V_Z}{Z}. \tag{1}$$

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 Fig. 2. Camera coordinate system is assumed to be coincided with the body coordinate system.

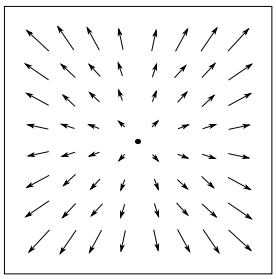


Figure 1: Divergence of optical flow vectors (flow divergence) when the observer is approaching a surface.

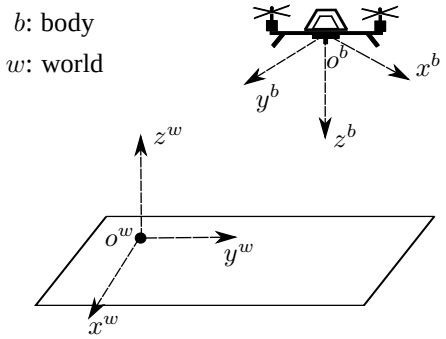


Figure 2: MAV body (x^b, y^b, z^b) and world (x^w, y^w, z^w) coordinate systems.

2.2 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 to a constant c, we can derive from Eq. 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 zeros when touching the ground.

A proportional feedback controller is used to track the desired flow divergence D^* or c as shown in Eq. (2):

$$\mu = K_p(D^* - D), \tag{2}$$

where K_p is the gain of the proportional controller.

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

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

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

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

It is clear that the model dynamics in Eq. (3) are linear but the observation in Eq. (4) is nonlinear. Fig. 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 descrease exponentially to zero in the end.

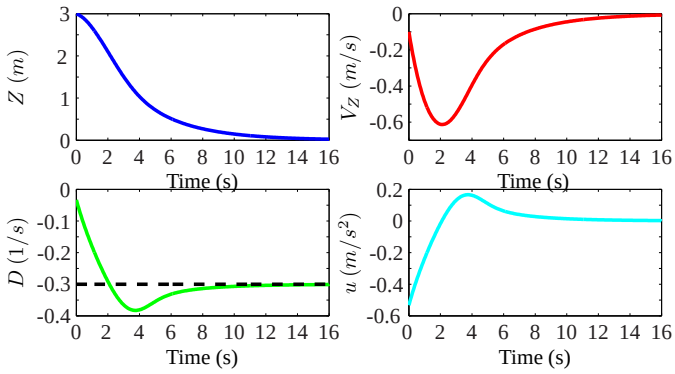


Figure 3: Height Z, velocity V_Z, and flow divergence D of an MAV during landing using constant flow divergence strategy ($D^* = -0.3\ s^{-1}$) with control input μ.

3 EKF-based height and velocity estimation and control using flow divergence

An overview of the methodology is presented in Fig. 4. Images captured from a camera are served as the inputs to the software while the control input commands the actuator to control the MAV. In this section, we describe: (1) the extended Kalman filter (EKF) algorithm using flow divergence and efference copies to estimate height and velocity of an MAV for landing control, (2) the vision method to compute flow divergence, and (3) the controller.

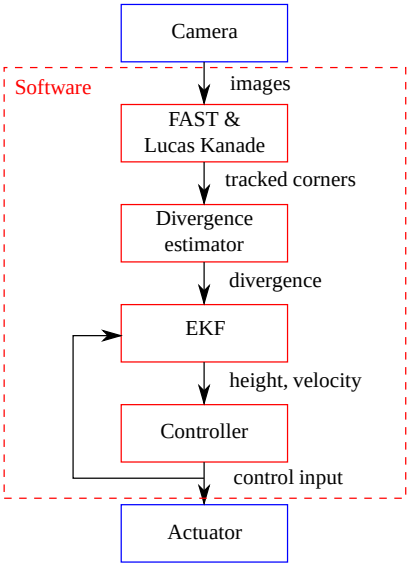


Figure 4: An overview of the methodology.

3.1 *Extended Kalman Filter*

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

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

$$\mathbf{z}_k = h\big(\mathbf{x}_k\big) + \mathbf{v}_k. \tag{6}$$

where $f(\cdot)$ and $h(\cdot)$ are the system matrix and the observation matrix which are obtained from Eq. 3 and 4. $\mathbf{w}(t)$ 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:

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

b) Covariance matrix of the state prediction error vector:

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

where $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are discretized matrices of $\mathbf{A}$ and $\mathbf{B}$, and can be computed as below:

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

c) Kalman gain:

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

where

$$\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}}\bigg|_{\hat{\mathbf{x}}_{k|k-1}} = \big[-\frac{\hat{x}_{2_{k|k-1}}}{\hat{x}^2_{1_{k|k-1}}}, \frac{1}{\hat{x}_{1_{k|k-1}}} \big]^T \tag{11}$$

d) Measurement update:

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

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:

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

3.2 *Features-based flow divergence estimation*

In computer simulation, we used Eq. 1 to compute flow divergence, while in flight tests, we estimate flow divergence using Eq. 14 (Proof can be found in [11].). For each image captured from an on-board camera, corners were detected using FAST algorithm [17, 18], and tracked in the next image using Lucas-Kanade tracker [19]. 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 take the arrange of these ratios and with known time interval between these two images Δt, we estimated divergence using Eq. 14:

$$\widehat{D} = \frac{1}{n} \cdot \frac{1}{\Delta t} \sum_{i=1}^{n} \big[\frac{d_{(t-\Delta t),i} - d_{t,i}}{d_{(t-\Delta t),i}}\big], \tag{14}$$

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.

3.3 *Controller*

Consider the common way using flow divergence to control the vertical dynamics of an MAV as shown in Eq. 2, it involves nonlinearity in the observation ($D = V_Z/Z$). In MAVs landings, the closer the MAV to the ground, the more sensitive the flow divergence is. Therefore, a nonlinear controller or a gain-scheduling method with the knowledge of initial height is needed to deal with the problem.

By using the proposed EKF algorithm, we are able to 'splits' the flow divergence into height and velocity. These two states can then be used to land MAVs using a linear PI controller as shown in Eq. 15.

$$\mu_k = K_{Z_1}(\hat{x}_{1,t_k} - x^*_{1,t_k}) + K_{Z_2} \sum_{i=1}^{k} (\hat{x}_{1,t_i} - x^*_{1,t_i}) \cdot \Delta t. \tag{15}$$

where K_{Z_1} and K_{Z_2} are the PI controller gains for tracking the desired height x^*_1. A desired landing profile can be chosen by assigning the values of x^*_1. Note that after obtaining

both height and velocity, it is possible to improve the control performance using gain optimization or a linear robust controller.

4 Computer Simulation

Before performing flight tests, we simulated the proposed algorithm presented in Subsection 3.1 in MATLAB to show the feasibility of the algorithm.

4.1 Landing simulation with simulated control inputs

In the simulation, we generated 3000 data (height and velocity) with a timestamp of $0.05\ s$ using a set of control inputs, μ as shown in Fig. 5e. These data were served as ground truth for validation. The flow divergence measurement was generated using Eq. 1 with measurement noise standard deviation of $0.001\ s^{-1}$ as illustrated in Fig. 5d.

By using the control input, μ and the flow divergence measurement, D, we estimated the height, Z and the velocity, V_Z with the proposed EKF algorithm. Fig. 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_{Z_0}) than the actual values. In addition, the innovation of EKF has zero mean meaning 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.

5 Experiments and Results

We implemented the EKF algorithm and vision algorithms presented in Section 3 in Paparazzi Autopilot, an open source autopilot software [20]. 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 two different control strategies for MAVs landings.

5.1 Landing using constant flow divergence

The first strategy we used in the flight tests is the constant flow divergence. A flow divergence setpoint was tracked for the whole landing. Instead of using P controller as described in Section 2, we used a PI controller to compensate the steady state error of the tracking. During the landing, the EKF algorithm was running in real-time to estimate the height and velocity of the MAV.

Fig. 6 shows the estimation results of the landing with a constant divergence of $-0.3\ s^{-1}$. The initial value of the height for EKF was set to be $\hat{Z}_0 = 3m$ which was different than the true initial height, i.e. $Z_0 \approx 2m$. In the figure, we can see that the height estimate converges to the true height,

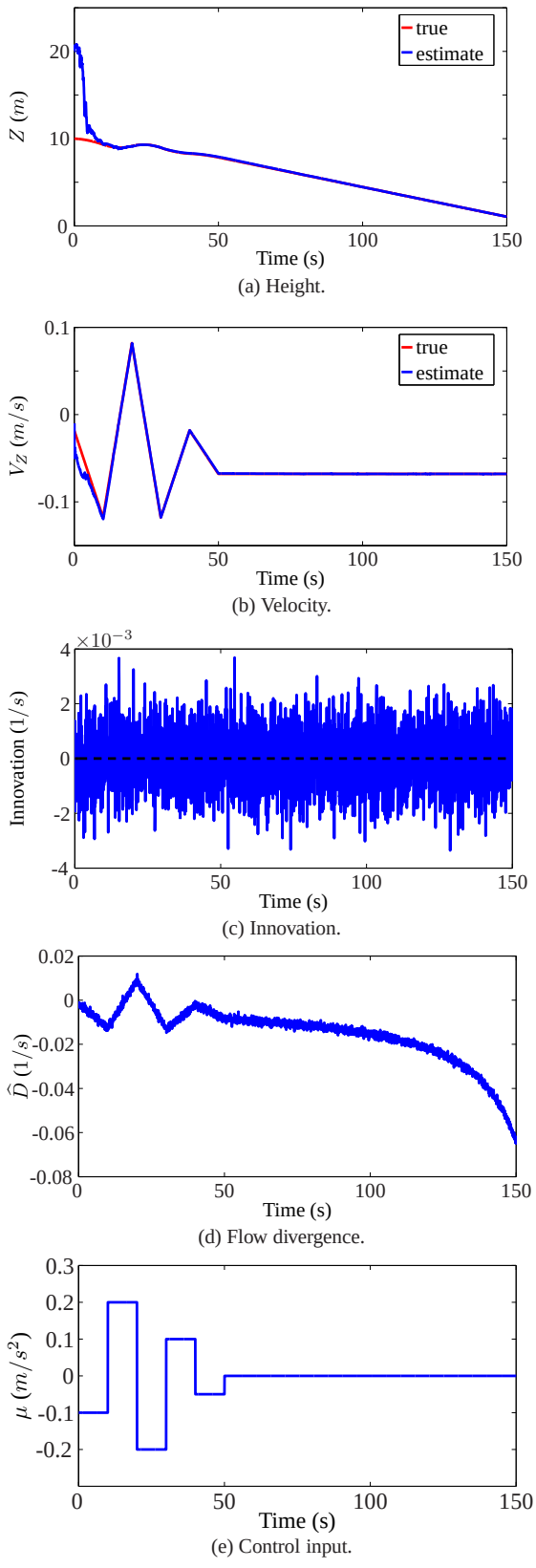


Figure 5: EKF-based height and vertical velocity estimation from flow divergence for landing control.

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 ($Z < 0.2m$) to allow the MAV for touchdown. Another fact is that the flow divergence ($D = V_Z/Z$) will become sensitive when Z is small. A fine-tuned control gain or a gain-scheduling method adapted with height is then needed to solve the sensitivity issue which might cause instability to the MAV. This issue can be seen from the measured flow divergence in Fig. 6d when the MAV is close to the ground.

To show the reliability of the proposed algorithm, we performed multiple landings with different flow divergence setpoints ($D^* = -0.1\ s^{-1}, -0.2\ s^{-1}, -0.3\ s^{-1}$) at different initial heights ($Z_0 \approx 2m, 3m$). The same initial guess of the height ($3m$) was used in the EKF algorithm for these flight tests. Fig. 7 shows the height and velocity estimates using EKF algorithm in different landings. In this figure, we can see that both height and velocity estimates match well with the true values. With a larger setpoint, 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.

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

5.2 Landing using height and velocity from EKF

The second 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 Fig. 8e 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 setpoints for landing.

Fig. 8 shows the height and velocity estimates from EKF compared with their ground truth, innovation of EKF, flow divergence from vision algorithms and from the ground truth, and the control input. After giving the initial excitation, the MAV started to track a height profile represented by the

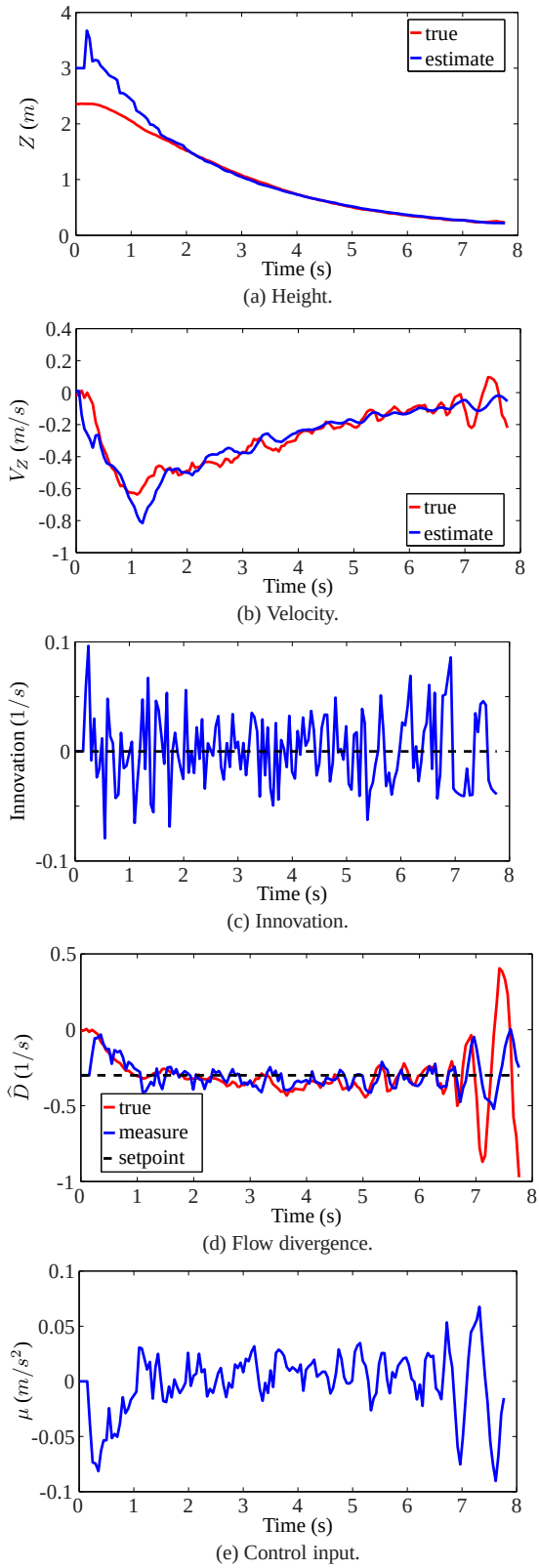


Figure 6: Landing with constant flow divergence ($D^* = -0.3\ s^{-1}$) with EKF-based height and velocity estimation.

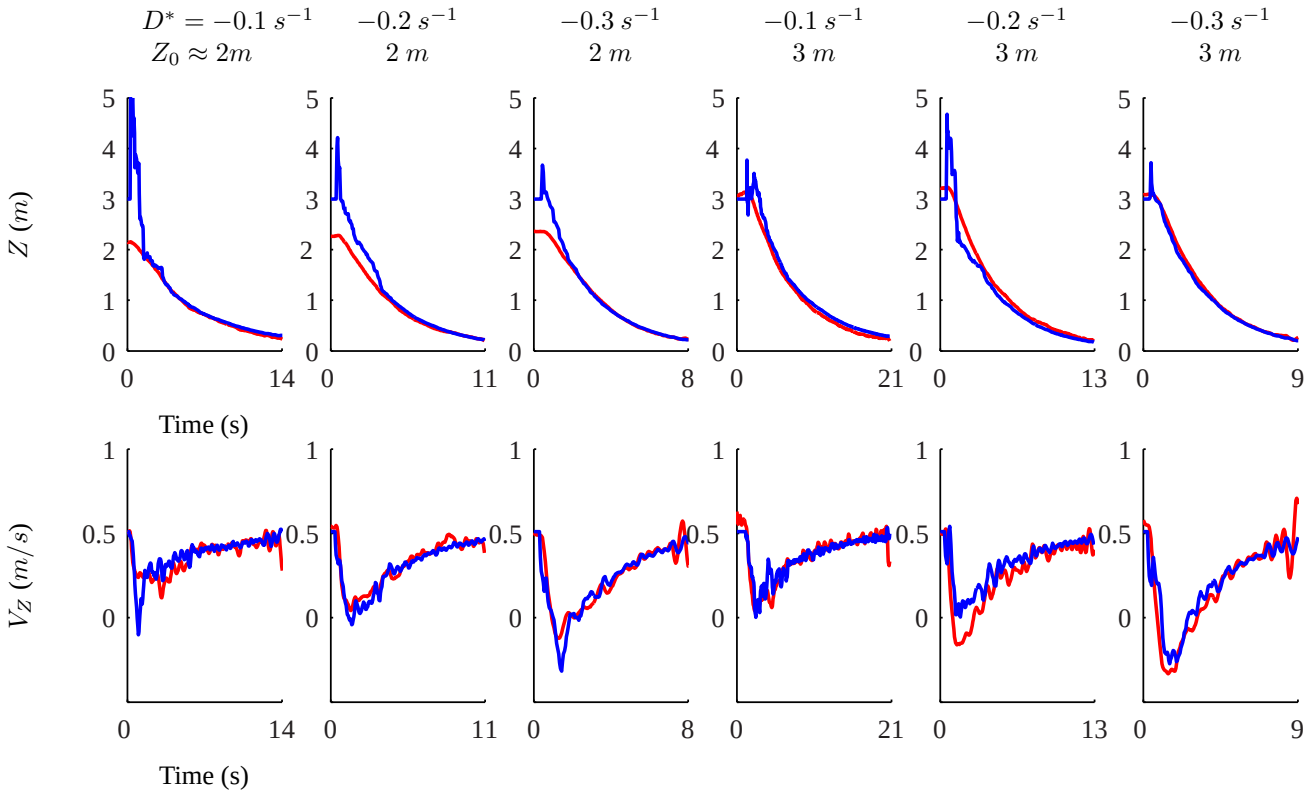


Figure 7: Multiple landings with different flow divergence setpoints ($D^* = -0.1\ s^{-1}, -0.2\ s^{-1}, -0.3\ s^{-1}$) at different initial heights ($Z_0 = 2m, 3m$). The estimates are shown with blue lines while their true values are represented by red lines.

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 $x^*_2 = -0.2ms^{-1}$. 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.

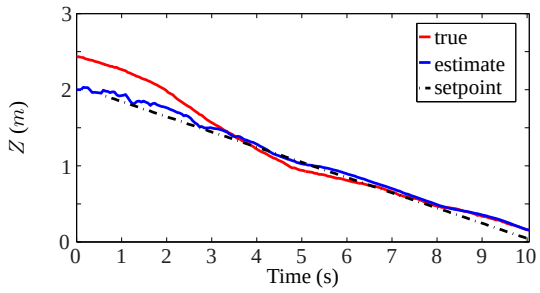
The second initialization was conducted in an outdoor flight by tracking a desired divergence for the first few seconds as shown in Fig. 9. Since *OptiTrack* system is not available for outdoor flight, we can only use the onboard sensor, such as Sonar, to provide the true height and velocity (dZ/dt) for accuracy comparison.

The landing started at $5\ m$ and the desired divergence of $-0.2\ s^{-1}$ was tracked. After $2\ s$, the controller was switched to track a height profile as presented in Fig. 9a which was generated based on the velocity estimate at the instance when the controller was switched ($\approx -0.8\ ms^{-1}$). In fact, the height profile can also be created based on a desired velocity (see Fig. 8). 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 Fig. 9 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 *OptiTrack* system while outdoor flight was conducted to show the robustness of the algorithm to the challenging outdoor environment. Some videos of the flight tests are available online [1].

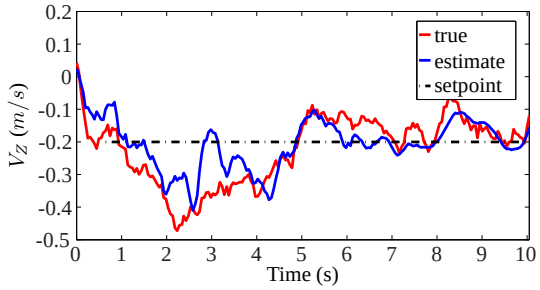
6 Conclusion

In conclusion, we proposed an algorithm to use an extended Kalman filter (EKF) to estimate height and vertical velocity of an MAV from flow divergence while approaching a surface, and use these estimates for landing control. This algorithm was tested in computer simulation as well as in flight tests. The results show that the proposed EKF approach managed to estimate accurately the height and velocity of the MAV compared to the ground truth provided by the external cameras. In addition, the estimates were used in the controller to 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 height and velocity which allows the use of the linear controllers. For future works, adaptive Kalman filter can be used to improve the estimation when vision data is hardly observable and strong external disturbances are involved. Also, the algorithm opens up the possibilities to use gain optimization techniques to improve the performance of the control.

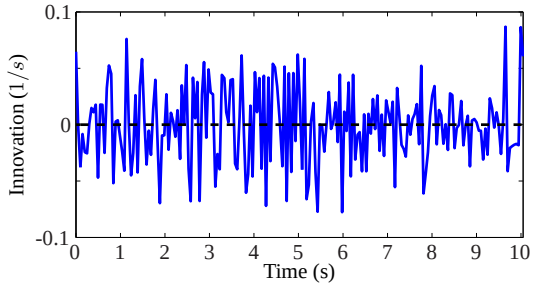
[1] Experiment videos: `https://goo.gl/KiJurr`



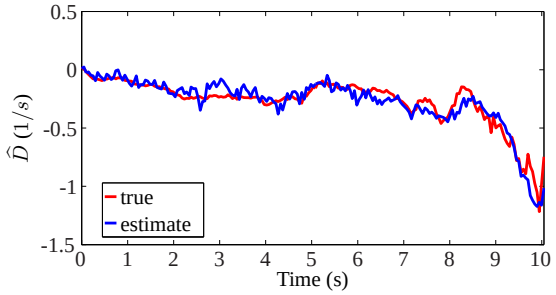
(a) Height.



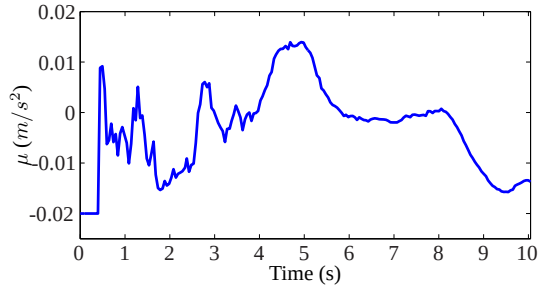
(b) Velocity.



(c) Innovation.

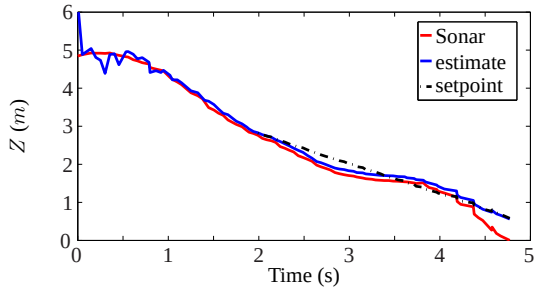


(d) Flow divergence.

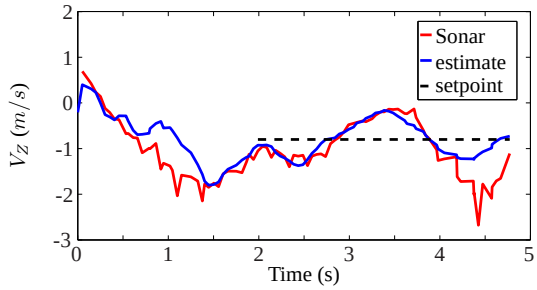


(e) Control input.

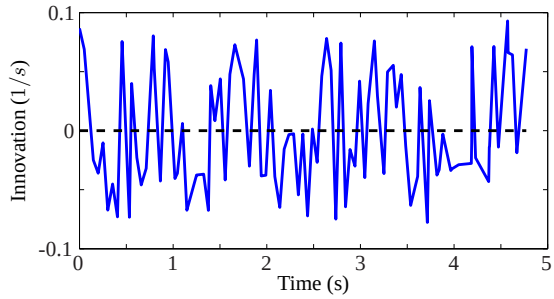
Figure 8: Landing with a linear controller using estimates from EKF (Indoor Flight).



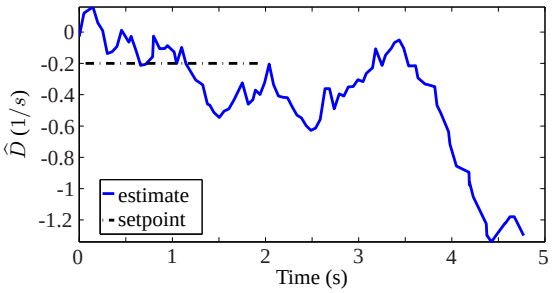
(a) Height.



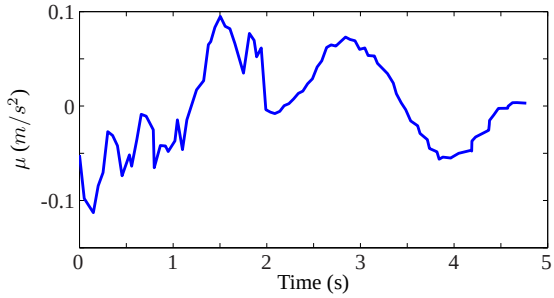
(b) Velocity.



(c) Innovation.



(d) Flow divergence.



(e) Control input.

Figure 9: Landing with a linear controller using estimates from EKF (Outdoor Flight).

REFERENCES

[1] C. de Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. de Croon. Autonomous flight of a 20-gram flapping wing MAV with a 4-gram onboard stereo vision system. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4982–4987. IEEE, 2014.

[2] G. C. H. E. de Croon, H. W. Ho, C. de Wagter, E. van Kampen, B. D. W. Remes, and Q. P. Chu. Optic-flow based slope estimation for autonomous landing. *International Journal of Micro Air Vehicles*, 5(4):287–298, 2013.

[3] M. Srinivasan, S. Zhang, M. Lehrer, and T. Collett. Honeybee navigation en route to the goal: visual flight control and odometry. *The Journal of Experimental Biology*, 199(1):237–244, 1996.

[4] L. F. Tammero and M. H. Dickinson. The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster. *Journal of Experimental Biology*, 205(3):327–343, 2002.

[5] W. E. Green and P. Y. Oh. Optic-flow-based collision avoidance. *Robotics & Automation Magazine, IEEE*, 15(1):96–103, 2008.

[6] C. McCarthy, N. Barnes, and R. Mahony. A robust docking strategy for a mobile robot using flow field divergence. *Robotics, IEEE Transactions on*, 24(4):832–842, 2008.

[7] B. Herisse, F.-X. Russotto, T. Hamel, and R. Mahony. Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 801–806. IEEE, 2008.

[8] F. Ruffier and N. Franceschini. Aerial robot piloted in steep relief by optic flow sensors. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1266–1273. IEEE, 2008.

[9] M. T Alkowatly, V. M Becerra, and W. Holderbaum. Bioinspired autonomous visual vertical control of a quadrotor unmanned aerial vehicle. *Journal of Guidance, Control, and Dynamics*, pages 1–14, 2014.

[10] H. W. Ho and G. C. H. E. de Croon. Characterization of flow field divergence for MAVs vertical control landing. In *AIAA Guidance, Navigation, and Control Conference*, page 0106, 2016.

[11] H. W. Ho, G. C. H. E. de Croon, E. van Kampen, Q. P. Chu, and M. Mulder. Adaptive control strategy for constant optical flow divergence landing. *ArXiv preprint arXiv:1609.06767*, September 2016.

[12] P. Bristeau, F. Callou, D. Vissiere, and N. Petit. The navigation and control technology inside the AR.Drone micro UAV. *IFAC Proceedings Volumes*, 44(1):1477–1484, 2011.

[13] J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[14] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 957–964. IEEE, 2012.

[15] F. van Breugel, K. Morgansen, and M. H. Dickinson. Monocular distance estimation from optic flow during active landing maneuvers. *Bioinspiration & biomimetics*, 9(2):025002, 2014.

[16] G. C. H. E. de Croon. Monocular distance estimation with optical flow maneuvers and efference copies: A stability-based strategy. *Bioinspiration & biomimetics*, 11(1):016004, 2016.

[17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[18] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[19] J. Bouquet. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs, http://www. intel. com/research/mrl/research/opencv*, 2000.

[20] B. D. W. Remes, D. Hensen, F. van Tienen, C. de Wagter, E. van der Horst, and G. C. H. E. de Croon. 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)*, pages 17–20, 2013.