Optical-flow-based Stabilization of Micro Air Vehicles Without Scaling Sensors

Braber, Titus; de Wagter, Christophe; de Croon, Guido; Babuska, Robert

Publication date
2018

Document Version
Final published version

Published in
10th International Micro-Air Vehicles Conference

Citation (APA)
Braber, T., de Wagter, C., de Croon, G., & Babuska, R. (2018). Optical-flow-based Stabilization of Micro Air Vehicles Without Scaling Sensors. In P. S. Watkins, & D. A. Mohamed (Eds.), *10th International Micro-Air Vehicles Conference: 22nd-23rd November 2018. Melbourne, Australia* (pp. 289-297)

Optical-flow-based Stabilization of Micro Air Vehicles Without Scaling Sensors

T.I. Braber*, C. De Wagter, G.C.H.E. de Croon, and R. Babuška
Delft University of Technology, Kluyverweg 1, 2629HS Delft, the Netherlands

ABSTRACT

This article presents an adaptive control strategy to stabilize a micro quadrotor in all three axes using only an Inertial Measurement Unit (IMU) for the attitude control and a monocular camera for canceling position drift. The proposed control scheme automatically determines the appropriate optical flow control gains. This is achieved by extending the stability-based approach to distance estimation developed in [1] to allow for the control of all three axes of a quadrotor. An analysis is done in simulation to present a proof of concept of the stabilization method and to determine the effects of scaling. Furthermore we verify the effects of varying effective camera frame rates and investigate how this control approach generalizes to smaller drone sizes. Actual flight tests are then performed on a Parrot ARDrone 2.0 and on a Parrot Bebop to show that both quadrotors achieve stable hover without position drift using only their IMU and bottom camera.

1 INTRODUCTION

Quadrotors are popular due to their simple structure and their combined capability of hovering and performing aggressive maneuvers. Being naturally unstable platforms, quadrotors require accurate and frequent estimations of their attitude, velocity and position. Attitude is obtained through onboard inertial measurements, but velocity and position require extra sensors. The most common outdoor solution is Global Positioning System (GPS) and the most common indoor solution is the use of a camera based external tracking system.

This has enabled impressive feats like pole [2] and ball [3] juggling quadrotors, quadrotors performing aggressive maneuvers [4], flying in swarms [5] and constructing structures [6]. However, in order to be of use in our daily lives, quadrotors also need to be able to fly in unknown and uncontrolled environments. While Simultaneous Localization And Mapping (SLAM) algorithms are widely used to solve these challenges, they are computationally expensive, which highly limits their applicability to smaller quadrotors. Therefore, several researchers are focusing on down-scalable solutions. The authors of [7] used a bottom facing camera combined with ultrasonic height sensor to estimate the quadrotor velocity. Various adaptations of the optic flow algorithms have been made to improve computational efficiency of computing ventral flow and divergence [8]. To counter long term drift in optic flow based velocity control, Li et al. [9] proposed a snapshot-based sensing and control method. To further reduce the required amount of sensors, the approaches in [1] and [10] use a bio-inspired detection of landing height solely based on optic flow divergence and IMU.

Autonomous quadrotors have become smaller with the recent advances in technology, allowing them to be used indoors while being inherently safer to operate around humans. Continuing to scale quadrotors down, however, is not evident as smaller mechanisms are more complex to produce, and parts such as motors cannot be scaled down indefinitely without losing performance. Furthermore the load carrying capability and onboard power of smaller quadrotors is limited, leading to restricted availability of sensors, computational power and flight time.

From [1] it is clear that controlling a quadrotor using only a minimal set of sensors, i.e. a monocular camera and an IMU is possible. This paper presents an extension of the stability-based approach of distance estimation to the control of all three axes. Secondly it presents the effects on the algorithm of downscaling a quadrotor.

Section 2 explains the control solution. In Section 3 we develop a simulation model and present the simulation results. Section 4 shows the results obtained in real flight tests. Finally, Section 5 concludes the paper.

2 STABILIZATION USING OPTICAL FLOW (OF)

Without high levels of intelligence to recognize known object sizes in scenes, the use of a single camera gives insufficient information to estimate depth directly. Although time-to-impact can be computed from OF directly, it is not able to differentiate between a small movement close to the scene, and a large movement further away from the scene as shown in Figure 1.

This difference is important for control, as in the latter case, the vehicle must slow down more. There are multiple solutions to solving this problem. As the OF is defined as the ratio between velocity over height, one option would be to simply measure the height using a second camera, or a range sensor. This is the most commonly used approach as most quadrotors have an on-board Ultrasonic Sensor (US) [7] or an Infrared (IR) range finder [8].

*Email address(es): TitusBraber@gmail.com

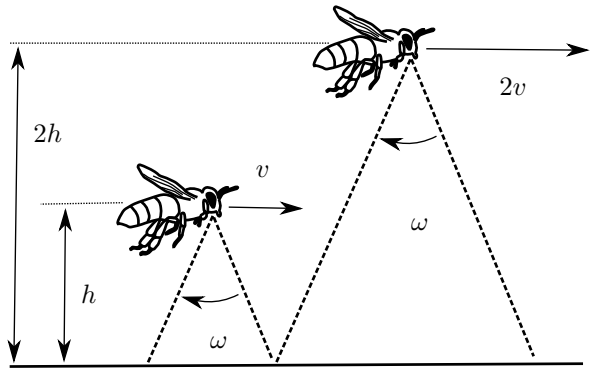


Figure 1: Optical flow does not provide scale.

Recently, different approaches have been proposed to scale optic flow without extra sensors. Given a sufficient acceleration, OF can be scaled based on accelerometers [11]. To control a quadrotor in hover, the need for constant acceleration is not ideal. Similarly, [1] showed that the motion of the quadrotor can be used to estimate the height (appendix B.2), though this also requires an actively moving quadrotor, which can be a problem in indoor environments. Alternatively, [1] also showed that the relationship between the control gain and the height for which the quadrotor becomes unstable due to self-induced oscillations can be used to estimate the height.

Following the proof in [1], there is a linear relationship between the height and the control gain for which the system becomes unstable. This instability leads to oscillations of the quadrotor as the phase shifts between control inputs and visual measurements.

Oscillations can be detected using the Fast Fourier Transform by looking for the specific frequency associated to the oscillations. However, as computational efficiency is of great importance, another method is proposed. Oscillations in the vertical direction can be detected by looking at the covariance of the divergence, which is the vertical speed divided by the height, and efference copies, which are the past thrust control inputs. Alternatively the auto covariance of the divergence can be used to detect a certain frequency of oscillation in the divergence itself, based on the auto covariance delay. However, when the quadrotor would go up and down as a result of an external disturbance such as a wind gust, an oscillation would show up in the auto covariance, as opposed to the covariance of the divergence and efference copies.

When detecting oscillations in the horizontal plane, the thrust is replaced by the effective thrust in the respective axis, or for the sake of simplicity the desired pitch or roll angle. The divergence is replaced by the ventral flow, which is the horizontal speed in X or Y axis divided by the height.

The relationship between gain and height can be used to estimate the current height by increasing the gain until oscillations are detected. With the estimated height, the proper control gains can be set for that height. This procedure can be seen in Algorithm 1 for the detection in the vertical axis.

Algorithm 1 Pseudo code for vertical loop PI control in hover

```
while true do
    if new div from vision module then
        error = −div
        if oscillating ≠ TRUE then
            increase gain: K_p += a · dt
            if Cov(div,thrust) > threshold then
                oscillating = TRUE
                Reduce K_p to stabilize: K_p = αK_p
            end if
        end if
        e_sum += error
        set thrust = K_p · error
    end if
end while
```

It is worth noting that this method is similar to the "the ultimate gain" PID tuning method by Ziegler and Nichols [12]. This empirical PID tuning method sets the PID gains to zero and increases the P (proportional) gain until the system becomes marginally stable, i.e. begins to oscillate. The P gain for which this happens is called the ultimate gain K_u and the period of the oscillation is T_u. These two values can be used to calculate the gains of the PID controller.

In the case of OF based control however, as the proper gains are height dependent due to the scale invariant measurements from OF, it is not sufficient to perform the Ziegler-Nichols tuning only once. Instead it is done adaptively while flying. Furthermore, the I gain is not set to zero during the period where the P gain is increased as the quadrotor can already substantially drift away with $K_i = 0$.

The resulting system not only tunes the OF based control loop, but at the same time scales the OF measurement.

3 Effects of Scaling Laws - Simulation Results

In this section the effects of scaling down a quadrotor will be investigated using simulation. An analysis of the scaling laws is used to properly scale down the Equations of Motion (EoM) used in the simulation. Finally, the effects of varying the Frames per Second (FPS) will be shown.

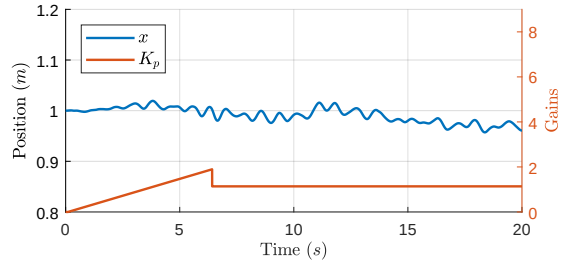
3.1 Simulation model

For simplicity and graphical representation, the simulation will be limited to 2 dimensions: the vertical Z dimension and the horizontal X dimension. The quadrotor is modeled as a rigid body with uniform mass, while the motors are modeled as discretized thrust forces with added Zero Mean White Noise (ZMWN). It is assumed that the maximum thrust the motors can deliver is twice the quadrotors weight. The EoM are as follows:

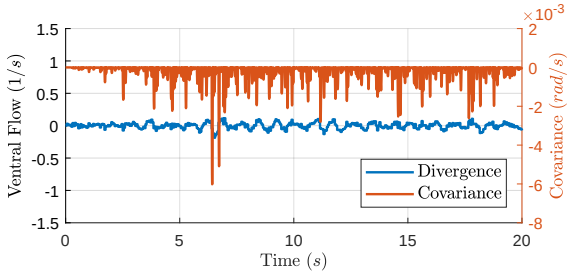
$$\ddot{x} = \frac{F}{m}\sin(\theta)$$
$$\ddot{z} = \frac{F}{m}\cos(\theta) - g$$
$$\ddot{\theta} = \frac{M}{I} \tag{1}$$

To simulate the OF measurement, the horizontal and vertical velocity will be taken divided by the height. This represents the ventral flow ($\dot{x}/z$) and divergence($\dot{z}/z$). The velocity and height measurements from the state are not known to the control system. While a typical camera has an update frequency of $30Hz$ or 30 FPS, the time spent computing the ventral flow and divergence might take longer than $1/30$ sec, resulting in a lower *effective* FPS. Therefore, to prevent unrealistic results, the vision signal will have to be discretized by a Zero Order Hold (ZOH) and delayed by a unit delay to model this effective FPS. During each simulation, the effective FPS is fixed, but it will be varied between simulations to study the effect of computational weight of vision algorithms. ZMWN is added to the vision signals before the ZOH, representing the estimation errors made by the vision algorithms.

In Figure 2a the position and the increasing gain in the horizontal axis can be seen with a similar algorithm to Algorithm 1 applied. Oscillations are detected using the covariance in Figure 2b, resulting in a stabilization of the gain.



(a) The position and gain of the quadrotor



(b) The height of the quadrotor for different scales

Figure 2: Results when **not** compensating for scaling

3.2 *Scaling analysis*

The starting point for the simulations will be a quadrotor based on the ARDrone 2, which is further defined as scale 1. This quadrotor will be scaled uniformly, which implies that all parts scale and their properties scale uniformly too. In reality these parts don't scale uniformly however. Instead, different types of motors, batteries, cameras, etc will have to be used, probably leading to differences not reflected in the simulations. The trend however should be visible.

Volume scales with a factor L^3. Mass therefore, assuming constant density, scales with L^3. Moment of inertia scales with a factor L^5, as $I = \int r^2 dm$ with $r \propto L$ and $m \propto L^3$.

The thrust that is generated by the motors, also has to be scaled. According to momentum theory or blade element theory, a thrust F can be approximated during hover in the following form:

$$F = 2\rho \cdot A \cdot v^2 \tag{2}$$

When scaling F in function of L, the density of air is unaffected $\rho \propto 1$, while the surface of the propeller scales with $A \propto L^2$. The rotor tip velocity v, scales differently depending on the assumption of compressibility of the flow[13].

Before making assumptions regarding rotor tip velocity, the effect on forces and moments thus far can be noted as

$$F \propto L^2 \cdot v^2$$
$$M \propto F \cdot L \propto L^3 \cdot v^2 \tag{3}$$

In the case of Mach scaling, the flow is assumed to be compressible and the rotor tip velocity to be constant, $v \propto 1$, as opposed to Froude scaling where the flow is assumed incompressible with a constant Froude number. Mach scaling is used here due to the compressibility of air, leading to:

$$F \propto L^2$$
$$M \propto L^3 \tag{4}$$

The effect of scaling on linear acceleration, $\ddot{x}, \ddot{z}$ with $F = m \cdot a$ and angular acceleration, $\ddot{\theta}$ with $M = I \cdot \alpha$ can be noted as

$$\ddot{x}, \ddot{z} \propto \frac{L^2}{L^3} \propto L^{-1}$$
$$\ddot{\theta} \propto \frac{L^3}{L^5} \propto L^{-2} \tag{5}$$

3.3 *Scaling compensation*

With the result of the previous section, Equation 1 can be updated in the following way:

$$\ddot{x} = \frac{F \cdot L^2}{m \cdot L^3}\sin(\theta) \quad = \frac{1}{L} \cdot \frac{F}{m}\sin(\theta)$$
$$\ddot{z} = \frac{F \cdot L^2}{m \cdot L^3}\cos(\theta) - g \quad = \frac{1}{L} \cdot \frac{F}{m}\cos(\theta) - g$$
$$\ddot{\theta} = \frac{M \cdot L^3}{I \cdot L^5} \quad = \frac{1}{L^2} \cdot \frac{M}{I} \tag{6}$$

It can be seen that smaller quadrotors have significantly faster dynamics due to the scaling in both linear and especially angular accelerations. To prevent the scaling effects from changing the behavior of the quadrotor, the control parameters can be scaled in such a way that they cancel the scaling effects where possible. Therefore the thrust F and moment M should be inversely scaled in comparison to Equation 4. To achieve this, the control parameters K_p, K_i, K_d should be scaled with L and L^2 respectively, looking at the control equation for the thrust and the moment respectively.

The scaling is summarized in Table 1. Please note that only the forces and moments will differ depending on the compensation.

Table 1: Scaling for physical quantities

Parameter	Symbol	Un-	Compensated
Width, Height	w, h	L^1	L^1
Volume, Mass	V, m	L^3	L^3
Moment of Inertia	I	L^5	L^5
Forces	F	L^2	L^3
Moments	M	L^3	L^5

For the simulations, the quadrotor will start from stable hover at $t = 0s$ when the simulation will be started. The control gains will then be increased until oscillations are detected. Once oscillations are detected, the best gain is known and selected, as will be explained further. The results from a simulation with $scale = 1$ form a base result and can be seen in Figure 3. It can be seen that the quadrotor controls its position and altitude for 20 seconds and only very slowly drift centimeters away from its original position.

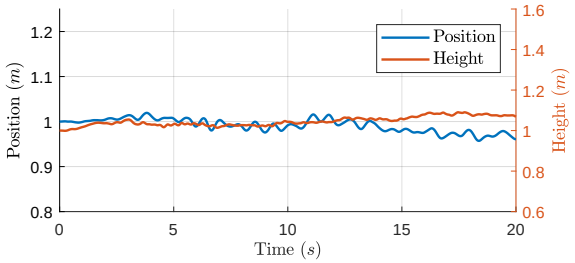


Figure 3: The position and height for $scale = 1$ in simulation

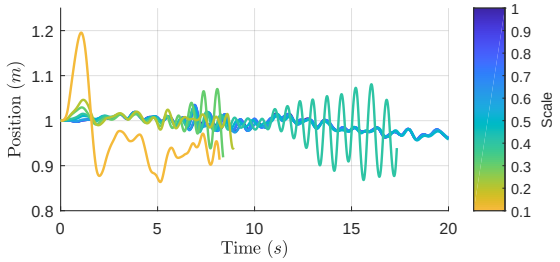
3.3.1 Uncompensated scaling

First, scaling will be applied to the quadrotor without the compensation in control gains mentioned in Section 3.3.

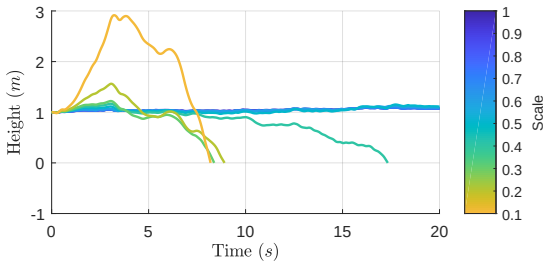
In Figure 4, the height and position of the quadrotor can be seen in the uncompensated case. The scale is varied from 1.0 to 0.1 in steps of 0.1 and the corresponding results are plotted in a color scale ranging from blue to yellow respectively.

Looking at the scales from 0.1 till 0.4 in Figure 4b, it can be seen that the smaller quadrotors become unstable and drift away increasingly as the size goes down. Note that the simulation stops for a scale when the quadrotor touches the ground. The cooler colors representing scales ranging from 0.5 to 1 show stable behavior.

Similarly, in the horizontal axis shown in Figure 4a, we can observe that the smaller drones start to oscillate in the X direction, whereas the larger quadrotors show the same stable behavior as in Figure 3.



(a) The position of the quadrotor for different scales



(b) The height of the quadrotor for different scales

Figure 4: Results when **not** compensating for scaling

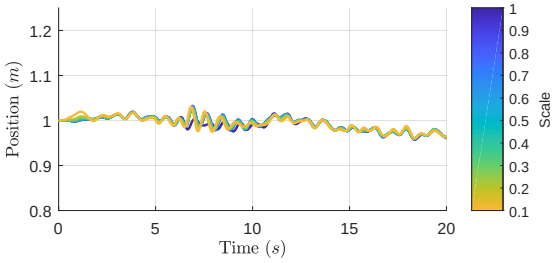
3.3.2 Compensated scaling

When the control forces and moments are scaled with L^3 and L^5 respectively (See Table 1), the plots in Figure 5 show a different result. It can be seen in both the horizontal and vertical axis that all scales are showing stable hover as in Figure 3.

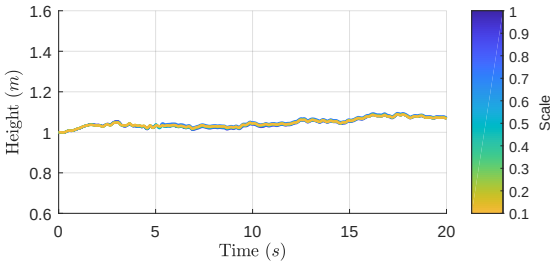
However, in Figure 5, the actuator noise from Section 3 has been scaled too. But in real life, noise tends to increase with smaller scale instead of decrease. To compensate for this increased sensor noise at smaller quadrotor scales, the sensor noise parameter is scaled back with L^2. This yields an important difference, which is shown in Figure 6.

For scale 0.1, the quadrotor goes left and right uncontrollably in the horizontal axis. While the other scales remain stable, there is nevertheless a difference with Figure 5 as the plots no longer overlap. Instead, they all follow the same trend with an increasing amplitude.

Figures 5 and 6 show that the control based on optic flow and IMU in principle can be scaled down to very small sized flying vehicles. The necessary condition, however, is that the

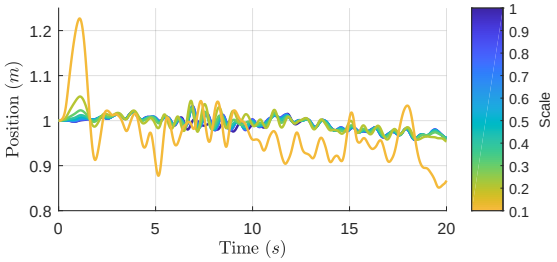


(a) The position of the quadrotor for different scales

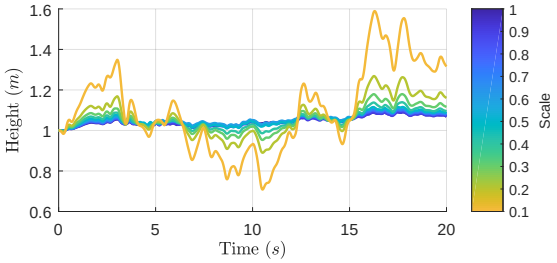


(b) The height of the quadrotor for different scales

Figure 5: Results when compensating for **both** scaling and noise



(a) The position of the quadrotor for different scales

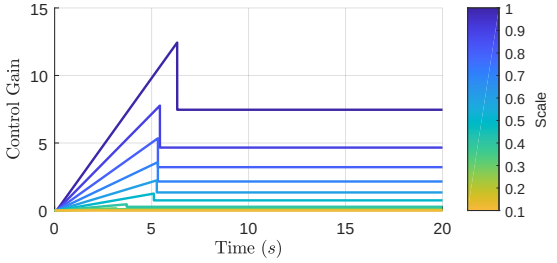


(b) The height of the quadrotor for different scales

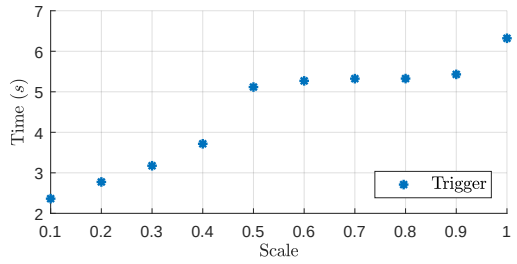
Figure 6: Results when compensating **only** for scaling, **not** for noise

sensor and actuator noise should reduce accordingly. Since in practice this is not the case, a minimal practical size exists.

As shown explained in detail in [1], the control gain of optic flow based control depends on the distance the observed surface. By slowly increasing the gain until oscillation is detected, the optimal gain is found. Figure 7 shows this gain



(a) The vertical gains of the quadrotor for different scales



(b) The vertical trigger points of the quadrotor for different scales

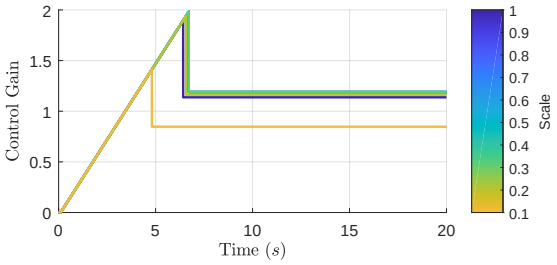
Figure 7: Vertical axis when compensating **only** for scaling, **not** for noise

increase and shows when the quadrotor detects a starting oscillation and selects the final gain. The increasing gains of the vertical axis are depicted in Figure 7a. The algorithm is triggered at roughly the same time for scales 0.9 through 0.5 between 5.4 to 5.1 seconds, while the gains from 0.4 to 0.1 trigger from 3.7 seconds to 2.4 seconds. Furthermore, scale 1 triggers at 6.3 seconds. Alternatively when plotting the scale against the trigger time, as Figure 7b shows, one could see a linear relation, with scales 0.9 to 0.5 breaking this trend.

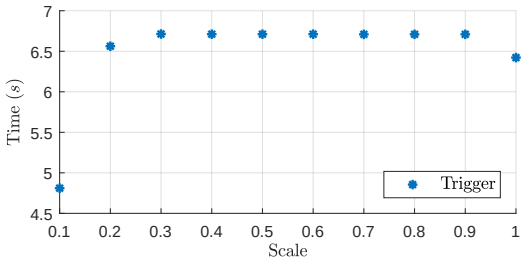
Looking at the horizontal axis in Figure 8 however, the triggers are all around the same time, 6.6 seconds, except for scale 0.1, which trigger at 4.8 seconds. This might again be explained by the fact that the actuator noise works in the thrust direction of the quadrotor, which is mainly vertical while hovering. The fact that all scales find roughly the same gain is because in the horizontal loop, the desired pitch angle θ is the actuator, which is in radians and thus not influenced by scaling.

3.4 Influence of varying effective FPS

The effect of reducing the FPS is analyzed using the model with scale$=1$. The quadrotor is set to hover in simulation at initial position in the horizontal axis of 1 meter and height 1 meter. Next the adaptive gain algorithm is started in both axes for different effective FPS. Figure 9 shows the response of the quadrotor in function of decreasing FPS. Below 15 FPS, the control of the quadrotor becomes unreliable and either starts to oscillate in the horizontal direction which also causes the drift in vertical direction or even becomes unstable at even lower frame rates.

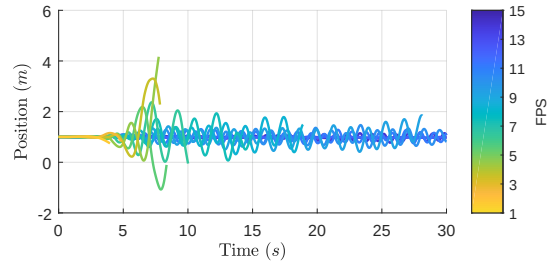


(a) The horizontal gains of the quadrotor for different scales

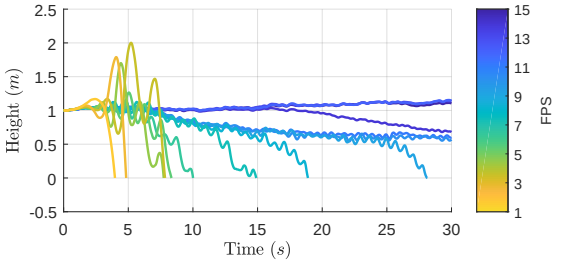


(b) The horizontal trigger points of the quadrotor for different scales

Figure 8: Horizontal axis when compensating **only** for scaling, **not** for noise



(a) The position of the quadrotor for different effective FPS



(b) The height of the quadrotor for different effective FPS

Figure 9: Determining the minimal FPS to fly a quadrotor of scale 1 using the adaptive gain strategy in both axis

We also investigate if it is possible to stabilize the quadrotor at smaller scales, as seen in Figure 6, with a higher effective FPS. The same experiment is performed, but for the smaller scales and higher effective FPS. Figure 10 shows the height of a 0.2 quadrotor, with an FPS scale from the previously used 20 FPS in yellow, to 100 FPS in blue, with steps

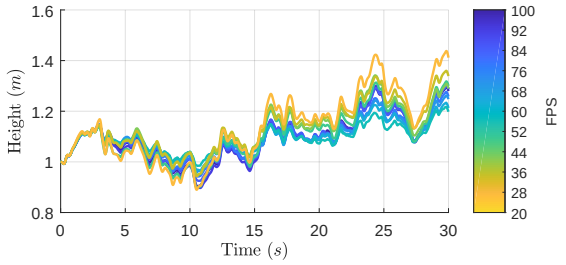


Figure 10: The height of a 0.2 scale quadrotor with increasing FPS

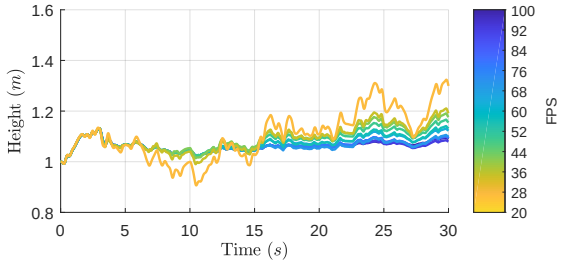


Figure 11: The height of a 0.2 scale quadrotor with increasing FPS without noise on the vision

of 8 FPS. Even though the quadrotor seems to benefit from a higher FPS, it is still not as stable as the 1 scale quadrotor was at 20 FPS. Furthermore, it seems there is an optimum, in this case 52 FPS. Figure 11 shows the same experiment, but without noise on the vision. Not only is this flight more stable than with noise, but now the increasing frame rate results in increasingly stable flight.

From this analysis, it can be seen that higher FPS only clearly helps if the noise is sufficient low, and that at lower scales even higher FPS can not nicely stabilize the quadrotor.

4 Flying a Quadrotor - Real Life Results

This section presents a complete control strategy that achieves stable hover in an unknown environment using just a single camera and an IMU. In the simulation results presented in Section 3 the algorithm was applied to both axes at the same time. With knowledge of the gain height relationships in all three axes however it is sufficient to use one axis to estimate the height and set the proper control gains for all three axes. This would require the quadrotor to apply Algorithm 1 in a single axis only, preferably the vertical one as it is the most stable one. Prior to this the linear gain height relationships should be determined for all axes.

4.1 Finding the gain height relationships

To determine the gain height relationships a quadrotor is flown in a controlled environment, the CyberZoo at the faculty of Aerospace Engineering at Delft University of Technology, where an OptiTrack motion capture system is installed to

acquire ground truth measurements. The quadrotor is set to hover and the algorithm is started in a single axis, while the other two axes are controlled using the ground truth measurements. When the quadrotor detects an oscillation it triggers to note the gain and the ground truth height. This experiment is repeated multiple times at several heights for all axes, allowing a linear fit through the data.

An ARDrone 2 flying with the open source Paparazzi software was used. The bottom camera acquired the ventral flows and divergence using the Lucas-Kanade (LK) algorithm at an effective FPS between 20 and 30. A picture of the experiment can be seen in Figure 12.

Figure 12: The ARDrone 2 during an experiment

It was assumed that the X and Y axes were symmetrical enough to determine the gain height relationship only in the Y axis. The results can be seen in Figure 13 for the horizontal axes and Figure 14 for the vertical axis.

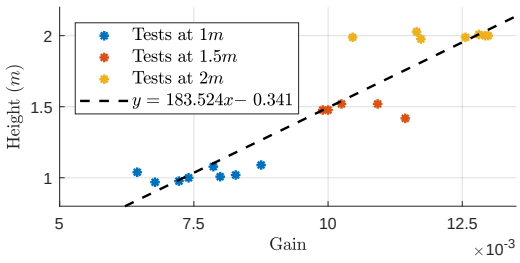


Figure 13: The gain and height for each trigger point in the horizontal axis

The fit for the horizontal axis shows a linear relationship for the gain and the height, at $y = 183.524x - 0.341$, and the vertical axis at $y = 0.995x + 0.066$.

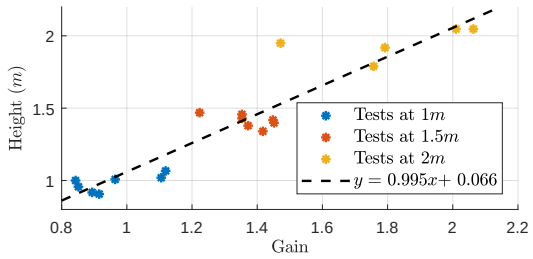


Figure 14: The gain and height for each trigger point in the vertical axis

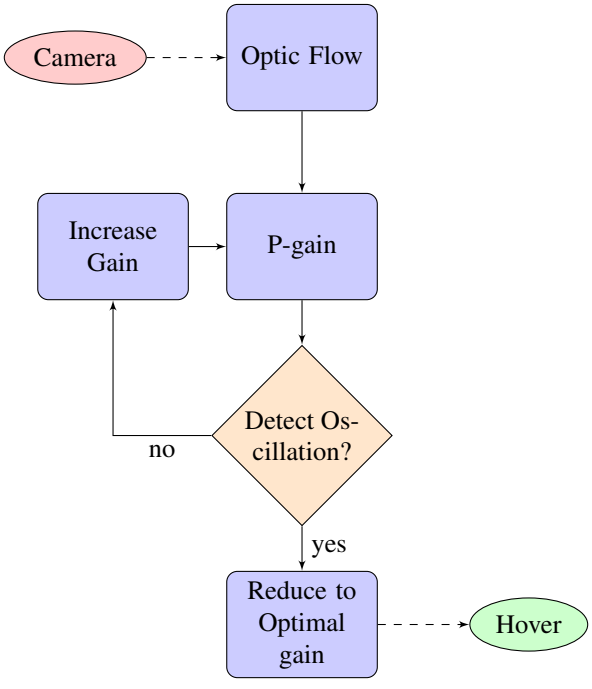
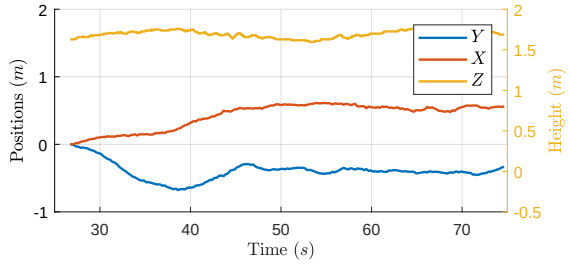


Figure 15: Flow chart of the optic flow based control

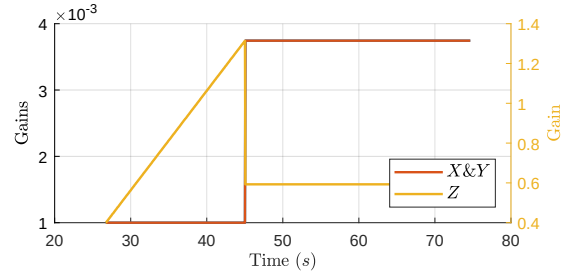
4.2 *Stable hover in an unknown environment*

To show the validity of this approach the ARDrone 2 is set to hover at an unknown height and the algorithm (See Figure 15) is started in the vertical axis, this can be seen in Figure 16b. Meanwhile the quadrotor is drifting away in the horizontal X and Y axes, as can be seen in Figure 16a. When the covariance of the divergence and thrust inputs dives below a threshold in Figure 16c the quadrotor detects the start of an oscillation it estimates the height it is flying at and and sets the proper gains in all three axes, using the gain height relationships as determined in Section 4.1.

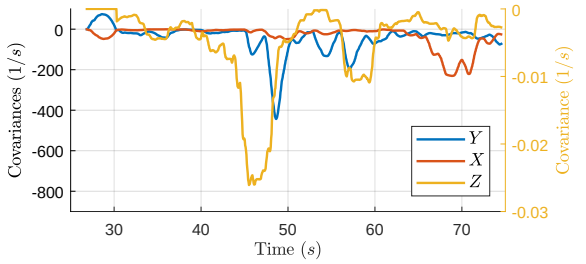
It is clearly visible that the trend of drifting in the horizontal direction is stopped and the quadrotor stabilizes around the position it was in when the proper gains were selected.



(a) The X and Y position and the height of the ARDrone 2



(b) The gains in X, Y and Z



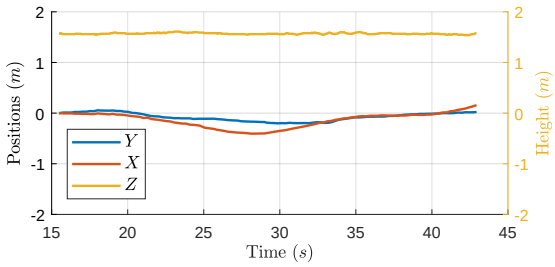
(c) The covariances used as a trigger

Figure 16: The ARDrone 2 with the algorithm applied to the vertical axis, to set proper gains in all axes using relationships between height and gain

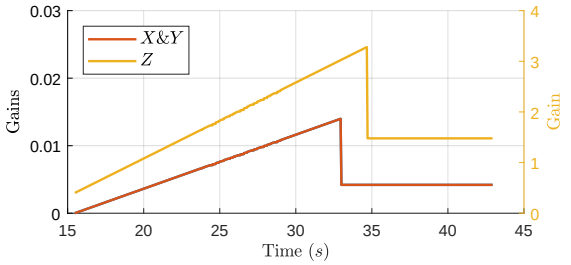
This experiment, amongst others, can also be seen in the following Youtube play list[1]. In video *Ardrone 2: Drift* the experiment mentioned can be seen, while video *Ardrone 2: drag* shows that the quadrotor stays above the texture mat, even if it is dragged around the Cyber Zoo.

Furthermore the video *Bebop: Simultaneous* shows an alternative approaches to the control strategy that will also work when no gain height relationship has been determined previously on the quadrotor. The video shows a Bebop quadrotor applying the algorithm on both the vertical axis and the horizontal axes at the same time. The same experiment is can be seen in Figure 17. When oscillations are detected in the vertical axis, the respective gain is lowered to stabilize that axis. When oscillations are detected in one of the horizontal axes, both gains are lowered to stabilize the horizontal axes.

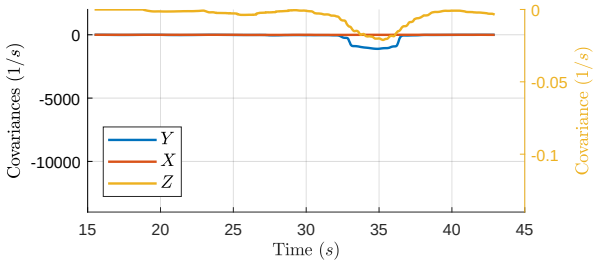
All code is released open-source in the Paparazzi-UAV project[2]



(a) The X and Y position and the height of the Bebop



(b) The gains in X, Y and Z



(c) The covariances used as a trigger

Figure 17: The Bebop with the algorithm applied to the all axes at the same time. A trigger in one horizontal axis will set the gain for both horizontal axes

[1] https://www.youtube.com/playlist?list=PL_KSX9GOn2P-Ire2SSVgxLnZorQ6jbgN1

[2] https://github.com/paparazzi/paparazzi/blob/master/conf/modules/optical_flow_hover.xml

5 Conclusion

In this article we have shown a novel control strategy to control a micro quadrotor drift free in an unknown environment using only an IMU and a monocular camera. The algorithm adaptively selects the proper control gains for the estimated height, using a stability-based approach to estimate distance. This was achieved by extending [1] to all axes and using the vertical height estimation to set control gains for both the vertical and horizontal axes from the predetermined gain height relationships. Alternatively the control strategy can also be used without predetermining these relationships, by applying the algorithm in each axes separately.

First the effects of scaling on stabilization of micro quadrotors have been shown by simulation. Though the dynamics are significantly faster for smaller quadrotors, especially the in the rotational degrees of freedom, it can be concluded that by proper scaling of the control gains most of the effects can be compensated for. The effects of noise however cannot be compensated for by inversely scaling control parameters when scaling down, leaving an increased effect of noise on the smaller quadrotors. It was also demonstrated that increasing the effective FPS would help decrease these effects on smaller quadrotors.

Finally it was successfully shown that quadrotors could hover without drifting away in both simulation and on an Parrot ARDrone 2 and Bebop using various versions of the control strategy.

Acknowledgements

This article is the condensation of my MSc thesis *Vision-based stabilization of micro quadrotors*, September 2017. I would like to thank my supervisors and fellow authors of this article again for their help and support during my thesis.

References

[1] Guido 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.

[2] Dario Brescianini, Markus Hehn, and Raffaello D'Andrea. Quadrocopter pole acrobatics. *IEEE International Conference on Intelligent Robots and Systems*, pages 3472–3479, 2013.

[3] Mark Müller, Sergei Lupashin, and Raffaello D'Andrea. Quadrocopter ball juggling. *IEEE International Conference on Intelligent Robots and Systems*, pages 5113–5120, 2011.

[4] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Springer Tracts in Advanced Robotics*, 79:361–373, 2014.

[5] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[6] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of Cubic Structures with Quadrotor Teams. *Mechanical Engineering*, 2011.

[7] Pierre-Jean Bristeau, FranÃ§ois Callou, David VissiÃ¨re, and Nicolas Petit. The navigation and control technology inside the ar.drone micro uav. *IFAC Proceedings Volumes*, 44(1):1477 – 1484, 2011. 18th IFAC World Congress.

[8] K N McGuire, G.C.H.E. de Croon, Christophe. De Wagter, K Remes B.D.W. Tuyls, and H J Kappen. Local Histogram Matching for Efficient Optical Flow Computation Applied to Velocity Estimation on Pocket Drones. *In press*, pages 1–15, 2015.

[9] Ping Li, Matthew Garratt, and Andrew Lambert. Monocular Snapshot-based Sensing and Control of Hover, Takeoff, and Landing for a Low-cost Quadrotor. *Journal of Field Robotics*, 32(7):984–1003, 2015.

[10] H. W. Ho, G. C. H. E. de Croon, E. van Kampen, Q. P. Chu, and M. Mulder. Adaptive gain control strategy for constant optical flow divergence landing. *IEEE Transactions on Robotics*, 34(2):508–516, April 2018.

[11] Volker Grabe, Heinrich H Bulthoff, and Paolo Robuffo Giordano. A comparison of scale estimation schemes for a quadrotor uav based on optical flow and imu measurements. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5193–5200. IEEE, 2013.

[12] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *InTech*, 42(6):94–100, 1995.

[13] Chester H Wolowicz and James S Bowman. NASA Technical Paper 1435 Similitude Requirements and Scaling Relationships as Applied to Model Testing NASA Technical Paper 14-35 Similitude Requirements and Scaling Relationships as Applied to Model Testing. (August):65, 1979.