

# Vision-based Altitude and Pitch Estimation for Ultra-light Indoor Microflyers

Antoine Beyeler, Claudio Mattiussi, Jean-Christophe Zufferey and Dario Floreano  
Ecole Polytechnique Fédérale de Lausanne (EPFL)  
Laboratory of Intelligent Systems  
CH-1015 Lausanne, Switzerland  
Email: name.surname@epfl.ch

**Abstract**—Autonomous control of ultra-light indoor microflyers is a difficult and largely unsolved task because of the strong limitations on the kind of sensors that can be embedded. We propose a new approach for altitude control of a 10-gram microflyer, where altitude as well as pitch angle are estimated using a set of visual, airspeed and gyroscopic sensors that weight about 1 (g) in total. This approach does not rely on an explicit estimation of optic flow, but rather takes as input the raw images as provided by the vision sensor.

We show that altitude and pitch angle of a simulated agent can be successfully estimated. This result is thus a first step toward autonomous altitude control of indoor flying robots.

## I. INTRODUCTION

Autonomous indoor flight poses a number of challenges that are yet to be solved. Unlike outdoor drones, the weight constraint precludes the use of many sensors. High precision inertial measurement units (IMU) are too heavy to be embedded on ultra-light microflyers, GPS signals are unavailable indoors and the horizon cannot be used as visual absolute angular reference. In general, active sensors, like distance sensors, tend to consume too much energy, as opposed to passive sensors such as simple CMOS cameras, MEMS rate gyros and anemometers. It is interesting to note that the fly seems to use the same kind of sensory modalities for navigation. The most important one is vision [1], [2], but gyroscopic information is also available thanks to organs called halteres [3], and it is probably the case for airspeed by the means of hairs and antennas [4].

It has been suggested that basic control of an indoor microflyer can be reduced to a minimal set of four behaviors<sup>1</sup>: *attitude control* (ATC), *course stabilization* (CS), *obstacle avoidance* (OA) and *altitude control* (ALC) [5], [6]. ATC consists of keeping the airplane roll and pitch angles stable. CS forces the airplane into straight or, at least, controlled trajectories when flying in free spaces, while OA ensures that it will not collide with walls and other obstacles. Finally, ALC keeps the airplane at proper altitude over ground or obstacles.

Implementation of the first three of these behaviors has already been demonstrated with a 30-gram airplane [7]. ATC was for the most part passively stabilized by the airplane

geometry, CS was implemented by a control loop that tied the rudder to a rate gyro, and OA was based on frontal optic flow (OF) divergence. However, in that study, no automatic ALC was implemented and an operator had to take care of it manually.

One possibility to address this problem is to take inspiration from insects. Bees have been shown to be acting as if they were regulating the OF perceived in the ventral part of their compound eyes [8]. Indeed, OF due to translation is inversely proportional to the distance over ground and proportional to the relative speed of the observer [9]. Therefore by maintaining OF constant the altitude can be regulated, given that the speed is known [10].

A few studies aiming at reproducing this mechanism based on OF or implementing it in flying robots have been completed, but they still suffer from severe limitations. Altitude control has been implemented on a gantry robot [8], and a tethered rotorcraft has been successfully controlled [11]. Both systems have a way to ensure that the OF detector is vertically oriented, downward pointing, which is not possible on a free-flying airplane. An OF detector that is fixed to an airplane in a perpendicular direction with respect to its axis will perceive a greater distance when the airplane has a nonzero pitch angle, thus the necessity to measure the pitch in order to either compensate for it, regulate it or actively control the OF detector optical axis. A simulated agent that successfully regulates altitude has been demonstrated [6], but both pitch and roll were controlled using color gradients present in the virtual environment. This cue is generally not available indoors. OF detectors have also been embedded on various autonomous aircrafts in order to control altitude [12]–[14], but in all cases the altitude controller ignores both the pitch angle and the pitch rate. The consequence of the later is a positive feedback<sup>2</sup> that makes the control unstable and forces the use of low gains in controllers, further reducing the control efficiency.

In this paper, we propose to reconsider the problem of vision-based altitude control. As a first step, we demonstrate that, contrary to previous studies, it is possible, using an optimization technique, to estimate altitude as well as pitch angle

<sup>1</sup>While not strictly necessary to control an microflyer in flight, *landing* and *take off* behaviors will also be necessary for practical applications.

<sup>2</sup>For example, a nose down pitch rate causes a reduction in the measured OF, thus an augmentation of the perceived altitude, which will lead the unaware controller to further apply a negative pitch rate command.

without preliminary OF estimation. The only required sensory signals are available on an existing 10-gram indoor microflyer called MC1 (Fig. 1), which is equipped with a camera, a rate gyro and an anemometer. Ultimately, such a method could be embedded in this microflyer to implement ALC and lead, using the other, already demonstrated behaviors, to a fully autonomous indoor microflyer.

The next section presents the theoretical ground of our method. The simulations we performed to assess the model and their analysis are presented in section III and IV. Finally, we discuss the results in section V.

## II. MODEL

Instead of first estimating OF and then estimating altitude and pitch from it, we propose to base altitude and pitch estimation on raw sensory data. For this first step toward a new ALC method, we simplify the problem by assuming that the microflyer has a null bank angle (the angle about the roll axis, i.e. the longitudinal axis of the airplane) at all times and that the projection of its trajectory on the ground is a straight line. This allows for reducing the problem to two dimensions, as represented in Fig. 2. Also, the ground is assumed to be planar and free of obstacles, and the air to be still.

Under these assumptions, altitude and pitch can be estimated by minimizing the difference between an interpolated image and an actual image, obtained by the sensor. This approach is similar to the image-interpolation technique proposed in [15], but instead of evaluating image motion we directly estimate the parameters we want to measure. Two images  $f$  and  $f'$  are grabbed from the vision sensor at time  $t$  and  $t + \Delta t$ . In the meanwhile, the other sensors are used to measure the microflyer velocities<sup>3</sup>  $v_x$  and  $v_y$ , and pitch rate  $\omega$ , that are assumed to be constant during the time interval  $\Delta t$ . Then, from  $f$ ,  $v_x$ ,  $v_y$ ,  $\omega$  and  $\Delta t$ , an interpolated image  $\hat{f}'(\theta, h)$  corresponding to  $f'$  and function of the altitude  $h$  and the pitch  $\theta$  is calculated. Finally,  $\theta$  and  $h$  are optimized so as to minimize the square error between the interpolated image  $\hat{f}'(\theta, h)$  and the actual image  $f'$ .

We model the vision sensor as a perfect, circular camera whose pixels sample the ground image at its intersection with their looking direction, noted  $\alpha_i$  for the  $i$ -th pixel (this angle has a negative value, since the camera points downward). The ground light intensity is noted  $I(x)$ , where  $x$  is the distance on ground calculated from the origin of the coordinate system, which is underneath the microflyer at time  $t$ . Then, the image  $f$  can be expressed as

$$f(\alpha_i) = I(l) = I(h \cdot \tan(\alpha_i + \theta + \frac{\pi}{2})) = I(h \cdot k(\alpha_i, \theta)), \quad (1)$$

where by definition  $k(\alpha, \theta) = \tan(\alpha + \theta + \pi/2) =$

<sup>3</sup>In principle, the anemometer provides only the speed along the airplane's main axis. The velocity  $v_y$  can be interpolated based on previous altitude estimations and used, together with the anemometer and previous pitch estimations, to estimate  $v_x$ .

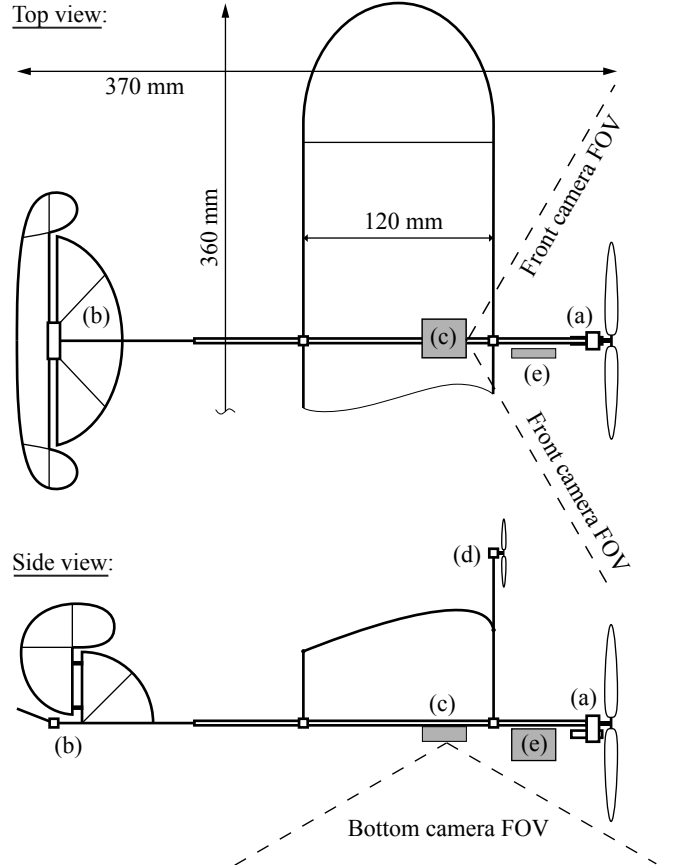
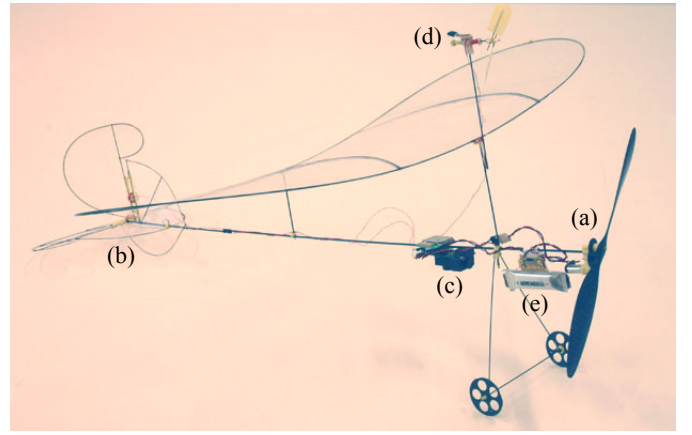


Fig. 1. Photograph and outline of the actual microflyer prototype, code named MC1. The MC1 is based on a microCeline, a 5.2-gram living room flyer produced by DIDEL (<http://www.didel.com>) that is equipped with a 4mm geared motor (a) and two magnet-in-a-coil actuators controlling the rudder and the elevator (b). When fitted with the required electronics for autonomous vision-based navigation, the total weight reaches 10 (g). The custom electronics consists of a microcontroller board (c) featuring a PIC18LF4620 running at 32MHz, a Bluetooth radio module (for parameter monitoring), and two camera modules, which comprise a gray-level CMOS linear camera (TSL3301) and a MEMS rate gyros (ADXRS150) each. One of those camera modules is oriented forward with its rate gyro measuring yaw rotations, and is meant to be used for obstacle avoidance. The second camera module is oriented downwards, looking longitudinally at the ground, while its rate gyro measures rotation about the pitch axis. Each of the cameras have 78 active pixels spanning a total field of view (FOV) of 120 (deg). In order to measure its airspeed, the MC1 is also equipped with an anemometer (d) consisting of a free propeller and a hall-effect sensor. This anemometer is placed in a region that is not blown by the main propeller (a). The 90 (mAh) Lithium-polymer battery (e) ensures an autonomy of approximately 15 minutes.

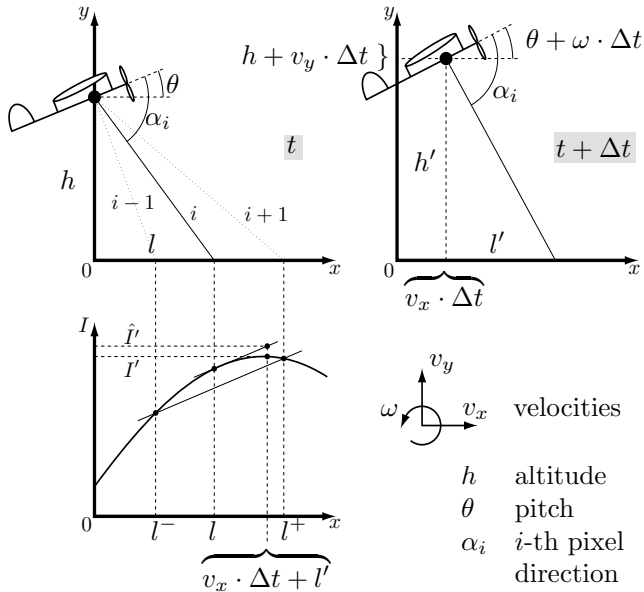


Fig. 2. Geometrical layout of the problem. The top graphs represent the airplane position in space at time  $t$  and  $t + \Delta t$ . The bottom graph shows how the ground texture intensity, noted  $I(x)$ , is interpolated, based on the intensity of the neighboring pixels.  $l$ ,  $l^+$  and  $l^-$  correspond to the position on ground sampled by pixels  $i$ , resp.  $i + 1$ , and  $i - 1$ . The local slope of the intensity function is estimated using  $I(l^-)$  and  $I(l^+)$ , and is used to interpolate  $I(v_x \cdot \Delta t + l')$  from  $I(l)$ , leading to the estimate intensity  $\hat{I}(v_x \cdot \Delta t + l')$  (see (3)).

$-1/\tan(\alpha + \theta)$ . Similarly, the second image is

$$\begin{aligned} f'(\alpha_i) &= I(v_x \cdot \Delta t + l') = \\ &= I(v_x \cdot \Delta t + (h + v_y \cdot \Delta t) \cdot k(\alpha_i, \theta + \omega \cdot \Delta t)). \end{aligned} \quad (2)$$

As represented in the bottom part of Fig. 2, we compute an interpolated image  $\hat{f}(\alpha_i)$  based on the linearization of  $I(x)$  around  $x = h \cdot k(\alpha_i, \theta)$ . Using the symbols defined in Fig. 2, we can write

$$\hat{I}(v_x \cdot \Delta t + l') = I(l) + \frac{I(l^+) - I(l^-)}{l^+ - l^-} \cdot (v_x \cdot \Delta t + l' - l). \quad (3)$$

Of course, this approximation is acceptable only under certain conditions. First, the interpolated point  $v_x \cdot \Delta t + l'$  should lie within the range  $[l^-; l^+]$  or close to it. This means that either the velocities (especially the rotational velocity) are limited, or the time interval  $\Delta t$  is kept short. Second, it limits the acceptable spatial frequencies for the ground texture, since the intensity should be close to linear in the range  $[l^-; l^+]$ . In practice, it is relatively easy to cut higher frequencies, for example by defocusing the vision system, but the image must contain some low frequencies for this method to be feasible.

Based on (1) and (2), we can rewrite (3), using for simplicity  $f(\alpha_i) = f_i$  and  $k(\alpha_i, \theta) = k_i(\theta)$

$$\begin{aligned} \hat{f}'_i &= f_i + \frac{f_{i+1} - f_{i-1}}{h \cdot (k_{i+1}(\theta) - k_{i-1}(\theta))} \cdot \\ &\cdot (v_x \cdot \Delta t + (h + v_y \cdot \Delta t) \cdot k_i(\theta + \omega \cdot \Delta t) - h \cdot k_i(\theta)). \end{aligned} \quad (4)$$

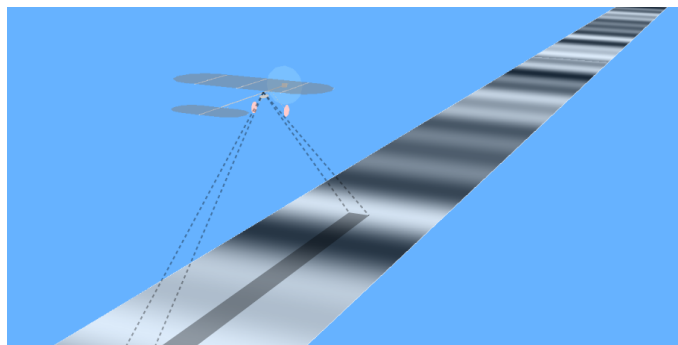


Fig. 3. The agent in the Webots simulator [16] with a representation of the field of view of the downward pointing camera.

Finally, we can write the following error function

$$\begin{aligned} \varepsilon(h, \theta) &= \sum_i (f'_i - \hat{f}'_i)^2 = \\ &= \sum_i \left[ f'_i - f_i - \frac{f_{i+1} - f_{i-1}}{k_{i+1}(\theta) - k_{i-1}(\theta)} \cdot \right. \\ &\cdot \left. \left( \frac{v_x \cdot \Delta t}{h} + \left( \frac{v_y \cdot \Delta t}{h} + 1 \right) \cdot k_i(\theta + \omega \Delta t) - k_i(\theta) \right) \right]^2. \end{aligned} \quad (5)$$

Minimizing  $\varepsilon(h, \theta)$  leads to an estimation of  $h$  and  $\theta$ .

### III. SIMULATIONS

To experimentally assess the model presented in the previous section, we use a flying agent simulated in the Webots simulator [16], as illustrated in Fig. 3. It is equipped with a downward-pointing linear camera whose geometry (field of view, number of pixels, etc.) is similar to that of the MC1 (Fig. 1). The agent can move in an artificial world composed of a textured ground. The ground texture is made of a sum of sines with frequencies ranging from 0 to 1 ( $m^{-1}$ ) and random phases.

In this paper, no attempt has been made to simulate the physics of a real microflyer since the goal is not to implement a control system for the microflyer. However, the velocities and trajectories imposed to the agent are kept within ranges that are reasonable for real indoor flyers like the MC1. In particular, the altitudes are kept between 0.5 and 2 (m), the pitch angle between  $-20$  and  $20$  (deg), the velocity between 1 and 2 (m/s) and the pitch rate below 20 (deg/s). The interval  $\Delta t$  is set to 5 (ms) in all simulations, to match what is technically feasible in terms of image acquisition frequency. All these numerical values have been derived from experimental data recordings from the real microflyer. While the agent is moved in the simulated world, all available data, including sensors, true positions and speeds, are logged for subsequent analysis. Finally, using the logged data for each time-step the error function in (5) is numerically minimized using Matlab's Optimization Toolbox to obtain an approximation of the altitude and pitch angle of the agent.

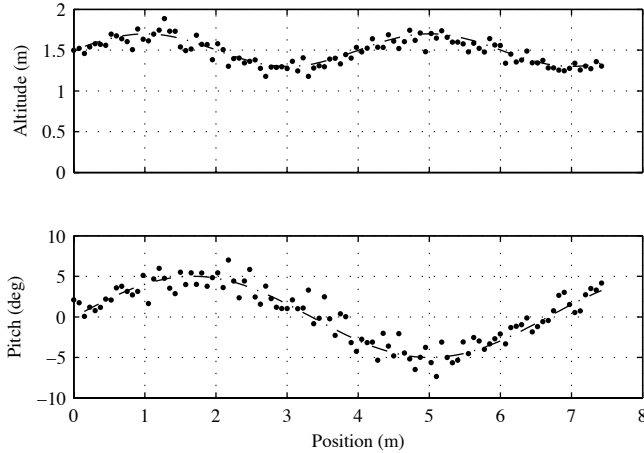


Fig. 4. Example of a trajectory and the corresponding estimation of altitude and pitch angle. The horizontal velocity  $v_x$  is set to 1.5, while both altitude and pitch angle are sinusoidal, by using proper, varying values for  $v_y$  and  $\omega$ . The true altitudes and pitches are marked by the dashed lines, while the circles represent the approximations.

#### IV. RESULTS

Fig. 4 shows true and estimated altitudes and pitch angles when the agent flies along a nontrivial trajectory. Here both vertical velocity and pitch rate have sinusoidal values over time, leading to sinusoidal trajectory and pitch angle. The graphs show that despite some variability, the estimations are on average very close to the actual values, even in a case where both pitch angle and pitch rate are nonzero.

To better characterize the estimation, a set of simple experiments are run. Fig. 5 compares the estimated altitude to the true altitude when the agent performs a level flight (a flight at constant altitude, i.e.  $v_y = 0$ ) with a constant forward velocity and a null pitch angle. The mean of the estimations stay within 1% of the true value up to an altitude of 1.5 (m). The variability tends to increase with the altitude. This is due to the fact that when the microflyer is higher, the sampling points of neighboring pixels are separated by a greater distance, therefore reducing the precision of the interpolation.

In the next set of experiments, the agent still performs a level flight, but at a constant, nonzero pitch angle. Fig. 6 shows that the pitch angle is, on average, estimated within 10% of the true value, up to angles of  $\pm 20$  (deg). Moreover, Fig. 7 shows that the average altitude estimation is not biased by the pitch angle. This is an interesting result showing that this method is capable of a correct estimation of altitude even when the airplane has a relatively large pitch angle, unlike in previous studies [12]–[14]. It must be noted, however, that both altitude and pitch estimations suffer from a slight increase in variability at high angles. This, again, is due to the fact that at high pitch values, some pixels are sampling the ground far in front of (or behind for negative pitch angles) the agent, leading to greater separation of sampling points and reduced precision of the interpolation.

In previous studies [12]–[14], OF generated by pitch rate was disturbing ALC even more than static pitch angle. Fig. 8

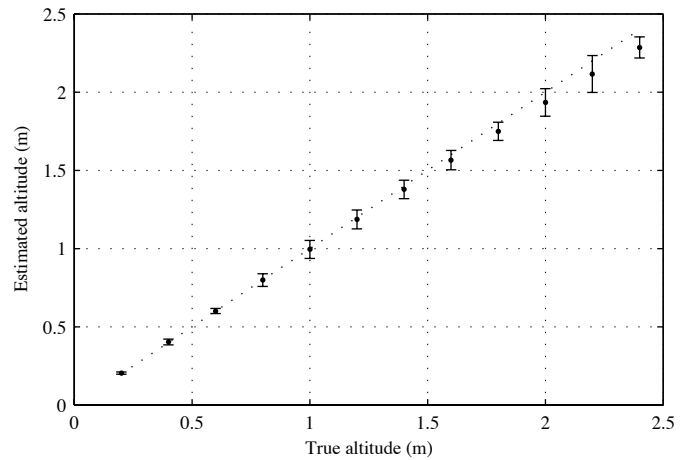


Fig. 5. Estimated altitude vs. true altitude, when the agent performs level flights at various altitudes. The velocity  $v_x$  is equal to 1.5 (m/s). The pitch was set to zero. Each data point corresponds to 200 estimations on a single level flight. The mean and the standard deviation of the estimation are shown.

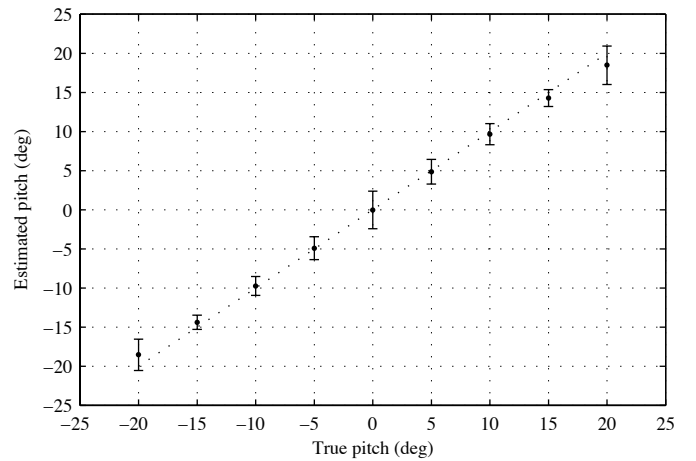


Fig. 6. Estimated pitch vs. true pitch, when the agent performs level flights with various pitch angles. The velocity  $v_x$  is equal to 1.5 (m/s). The altitude is fixed to 1 (m). Each data point corresponds to 200 estimations on a single level flight. The mean and the standard deviation of the estimation are shown.

shows that with our method, altitude estimation is not biased by nonzero pitch rate, remaining on average well below a 1% error within  $\pm 20$  (deg/s). The variability of the measurement is not affected either. As an example, by not compensating for a pitch rate of 20 (deg/s) an OF detector would see, in similar conditions, an augmentation of the OF in the order of 25%, leading to an altitude estimation 20% below the true value. Such a bias makes altitude control intrinsically unstable, because an unaware controller would further increase the pitch to catch up with altitude, leading to a positive feedback loop.

To summarize, these results show that the estimation of altitude using the method we propose does not suffer from significant biases, even in the cases where the agent has nonzero pitch angle and pitch rate. Of course, there is some variability in the estimation since the first-order interpolation is not exact, but it is easy to cope with this problem using

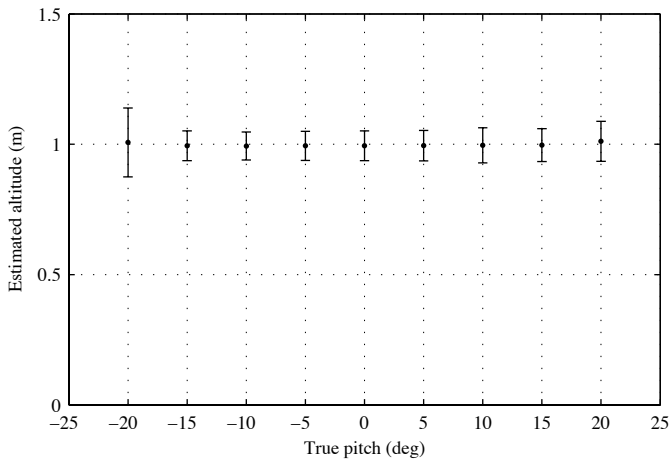


Fig. 7. Estimated altitude vs. true pitch, when the agent performs level flights with various pitch angles. The velocity  $v_x$  is equal to 1.5 (m/s). The altitude is fixed to 1 (m). Each data point corresponds to 200 estimations on a single level flight. The mean and the standard deviation of the estimation are shown.

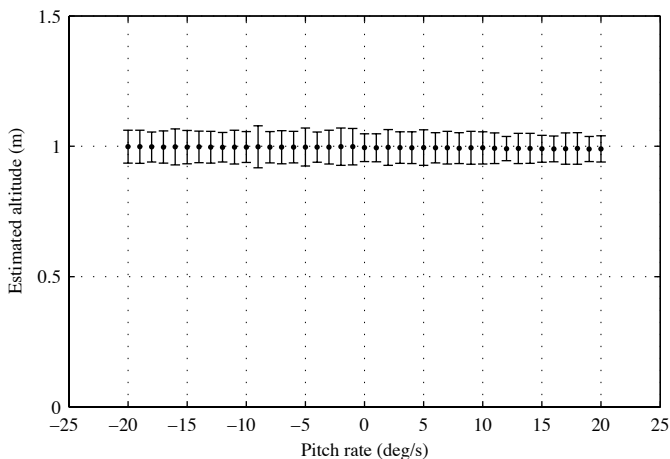


Fig. 8. Estimated altitude vs. pitch rate, when the agent performs level flights with various pitch rates. The velocity  $v_x$  is equal to 1.5 (m/s). The altitude is fixed to 1 (m) and the pitch angle kept in the range  $[-20; 20]$ . Each data point corresponds to 200 estimations on a single level flight and the error bars show the standard deviation on the measurements.

temporal low-pass filtering on the estimation signal. Moreover, the blurring due to defocused optics could help to cut the apparent high frequencies seen by the pixels pointing far in front or behind the microflyer<sup>4</sup>.

## V. DISCUSSION AND FUTURE WORK

We have demonstrated that it is possible to reliably estimate altitude and pitch of an microflyer using the raw data provided by simple vision, inertial and airspeed sensors that have already been embedded in a 10-gram indoor flying robot. This is achieved without any distance sensors or IMUs that are generally too heavy and consume too much power to be of practical use on such platforms.

<sup>4</sup>The pixels pointing far in front of the plane sample the ground at a greater distance from each other. The ground texture has then a comparatively higher frequency.

Of course, several limitations need to be overcome to actually use this technique to control the altitude of a real robot. For this initial study, we simplified the model in ways that are not practical for the real microflyer. First, the airplane is assumed to have a null bank at all times, which is obviously not the case in reality. To cope with this problem, the model could be extended to three dimensions, considering not only the pitch angle and rate, but also the roll angle and rate. This would require a vision system equipped with a 2D sensor. Recent technological progresses make us confident that it will soon be possible to embed 2D vision systems in a 10-gram microflyer (VGA camera modules that weight less than 0.5 (g) are already commercially available). Alternatively, the controller could be made to estimate altitude only when the microflyer is known to have no roll angle and rate. For instance, in [7], the airplane is forced into straight trajectories during which obstacles are detected and avoidance takes the form of a short, open-loop saccade. A similar approach could be used where ATC would be active only during straight flight.

The used camera model is also too simple to faithfully represent a real vision system. In this model, pixels sample the ground texture at the intersection of their looking direction. In a real camera, the value of each pixel is the convolution of the image with a kernel and corresponds to a finite, nonzero field of view, especially when the optics are defocused to cut disturbing high frequencies. For insect eyes, this kernel has been shown to be approximately gaussian [17]. In the future, we will extend the model to take this property into account. This should allow us to reduce the requirements for the ground texture and eventually assess the method with real images by means of, for example, an actuated rotating arm or directly with the MC1 along with a tracking system.

The most important limitation is the required computational power. Each estimation requires the minimization of a rather complicated nonlinear function of two variables. This cannot be practically implemented on the 8-bit microcontroller embedded in the MC1. A way of simplifying this computation must be found. A possibility that we will investigate in the future is the use of neural networks that would be trained off-line using learning or genetic algorithms. This could be made possible by feeding to the network higher level primitives inspired by the various terms found in (5). An added advantage of this approach would be that it could be extended to directly handle the control of the microflyer altitude without explicitly evaluating it, using the information that we demonstrated to be present in the sensory signals.

A clear advantage of the method we demonstrated is that it does not require any preliminary OF computation, thus avoiding the need to choose a particular estimation method among those available (Elementary Motion Detectors [18], image-interpolation techniques [15], or others [19], [20]). Also, there is no need to select a particular direction for the OF detectors [12]–[14] or to use matched filters [6]. The geometry of the vision system (i.e. the pixel looking directions) is automatically taken into account by the model and used for the estimation. This allows for a reliable estimation even with

nonzero pitch angle or pitch rate, which is a clear improvement compared to previous studies.

#### ACKNOWLEDGMENTS

We wish to thank Jean-Daniel Nicoud for his invaluable collaboration in building the MC1. We are also grateful to Adam Klaptocz who has proofread the manuscript. This project is supported by the Swiss National Science Foundation, grant 200021-105545/1.

#### REFERENCES

- [1] M. Egelhaaf and A. Borst, "A look into the cockpit of the fly: Visual orientation, algorithms, and identified neurons," *The Journal of Neuroscience*, vol. 13, no. 11, pp. 4563–4574, 1993.
- [2] ———, "Motion computation and visual orientation in flies," *Comparative biochemistry and physiology*, vol. 104A, no. 4, pp. 659–673, 1993.
- [3] G. Nalbach and R. Hengstenberg, "The halteres of the blowfly calliphora - three-dimensional organization of compensatory reactions to real and simulated rotations," *Journal of Comparative Physiology A*, vol. 175, pp. 695–708, 1994.
- [4] R. Dudley, *The Biomechanics of Insect Flight: Form, Function, Evolution*. Princeton University Press, 2000.
- [5] J.-C. Zufferey, A. Beyeler, and D. Floreano, "Vision-based navigation from wheels to wings," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2003, pp. 2968–2973.
- [6] T. Neumann and H. Bülthoff, "Behavior-oriented vision for biomimetic flight control," in *Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics*, 2002, pp. 196–203.
- [7] J.-C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions on Robotics*, vol. 22, no. 1, 2006.
- [8] M. Srinivasan, S. Zhang, J. Chahl, E. Barth, and S. Venkatesh, "How honeybees make grazing landings on flat surfaces," *Biological Cybernetics*, vol. 83, pp. 171–183, 2000.
- [9] J. Koenderink and A. van Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, pp. 247–254, 1987.
- [10] F. Mura and N. Franceschini, "Visual control of altitude and speed in a flying agent," in *From Animals to Animats III*. MIT Press, 1994, pp. 91–99.
- [11] F. Ruffier and N. Franceschini, "Optic flow regulation: the key to aircraft automatic guidance," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 177–194, 2005.
- [12] G. Barrows, C. Neely, and K. Miller, "Optic flow sensors for MAV navigation," in *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, ser. Progress in Astronautics and Aeronautics, T. J. Mueller, Ed. AIAA, 2001, vol. 195, pp. 557–574.
- [13] W. Green, P. Oh, K. Sevcik, and G. Barrows, "Autonomous landing for indoor flying robots using optic flow," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 2, 2003, pp. 1347–1352.
- [14] J. Chahl, M. Srinivasan, and H. Zhang, "Landing strategies in honeybees and applications to uninhabited airborne vehicles," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 101–110, 2004.
- [15] M. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biological Cybernetics*, vol. 71, pp. 401–416, 1994.
- [16] O. Michel, "Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1(1), pp. 39–42, 2004.
- [17] M. Land, "Visual acuity in insects," *Annual Review of Entomology*, vol. 42, pp. 147–177, 1997.
- [18] W. Reichardt, "Movement perception in insects," in *Processing of optical data by organisms and by machines*, W. Reichardt, Ed. New York: Academic Press, 1969, pp. 465–493.
- [19] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [20] A. Verri, M. Straforini, and V. Torre, "Computational aspects of motion perception in natural and artificial vision systems," *Philosophical Transactions of the Royal Society B*, vol. 337, pp. 429–443, 1992.