Autonomous flight at low altitude with vision-based collision avoidance and GPS-based path following

Jean-Christophe Zufferey, Antoine Beyeler and Dario Floreano

***Abstract*— The ability to fly at low altitude while actively avoiding collisions with the terrain and other objects is a great challenge for small unmanned aircraft. This paper builds on top of a control strategy called *optiPilot* whereby a series of optic-flow detectors pointed at divergent viewing directions around the aircraft main axis are linearly combined into roll and pitch commands using two sets of weights. This control strategy already proved successful at controlling flight and avoiding collisions in reactive navigation experiments. This paper shows how *opitPilot* can be coupled with a GPS in order to provide goal-directed, nap-of-the-earth flight control in presence of static obstacles. Two fully autonomous flights of 25 minutes each are described where a 400-gram unmanned aircraft is flying at approx. 9 m above the terrain on a circular path including two copses of trees requiring efficient collision avoidance actions.**

I. Introduction

A small unmanned aircraft flying at low altitude while avoiding collisions with objects on the ground is not only interesting for military applications, but could be of great help in many civilian tasks as well. One can think of ultra-low-altitude imagery to construct 2D or 3D maps with unprecedented resolution and realism; measuring air quality in urban environments to better understand pollution spreading and alert the population only when and where necessary; measuring radio signal strength in order to determine the coverage of mobile telephony antenna or network access points; search for lost people; transport small parcels across a city; etc. Commercially available miniature autopilots solve the problem of flight stabilization and way-point navigation in free spaces using GPS+IMU[1], but they offer no practical solution to cope with obstacles.

On the research side, we find two kinds of approaches attempting at solving this problem. The first one consists of relying on classical GPS+IMU autopilots [1] and to add sensors that scans the environment in order to feedback path corrections into the autopilot, which is at the core of the navigation process [2], [3], [4]. These methods tend to be heavy and computationally intensive because a 3D map of the environment needs to be maintained in real-time and the sensors used to perceive depth need to be highly accurate in order for the algorithms to converge. Here we propose an alternative approach that consists of solving the collision-avoidance problem even before adding goal-directed navigation ability on top. Following the behaviour-based philosophy [5], the idea is to develop a system that can wander around without hitting objects before adding a navigation layer on top of it.

In order to ensure the low-level collision-avoidance capability, some bio-inspired researchers have looked at using optic flow as a parsimonious way of perceiving depth around an aircraft and reactively maintain a safe altitude over ground [6], [7], avoid lateral collisions [8], [9], or completely control the 3D behaviour of an aircraft [10], [11], [12]. In particular, we recently proposed a control strategy called *optiPilot* [13] that maps optic flow as sampled in many divergent viewing directions and map them into steering commands. *OptiPilot* requires neither GPS nor IMU to control the flight and avoid collisions. However, in all of these contributions, the flying system was passively guided by the layout of the environment or physically constrained to some circular paths either by the use of a tether or a closed arena.

For the first time, this paper shows how a simple low-level 3D collision avoidance strategy can be coupled to GPS-defined path following requiring neither mapping of the environment, nor computationally intensive algorithms, nor any explicit measurement of the attitude of the aircraft. To begin with, the next section provides an overview of the low-level *optiPilot* control strategy as well as its coupling with a higher-level GPS-based path following algorithm. The third section describes the flying platform and sensors that were used to test our approach in natural environments. Finally the flight data acquired during a fully autonomous flight are presented to show in more detail the functioning of this control strategy.

II. Control strategy

OptiPilot is a way of mapping a series of optic flow signals sampled in a set of divergent viewing directions into roll and pitch rate commands (see Fig. 1 for a reminder on the aeronautical axes of an aircraft). This control strategy couples the fundamental properties of optic flow [15] with the motion constraints of translating aircraft [14] to turn linear combinations of optic-flow data into flight commands. In this section, we first provide a summary of this control strategy before describing how the GPS position of the

This work has been supported by the Swiss National Science Foundation (grant 200020-116149).

J.-C. Zufferey is with Laboratory of Intelligent Systems (http://lis.epfl.ch), Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland jean-christophe.zufferey@epfl.ch

A. Beyeler is CTO of *senseFly* (http://www.sensefly.com), 1024 Ecublens, Switzerland antoine.beyeler@sensefly.com

D. Floreano is the head of the Laboratory of Intelligent Systems (http://lis.epfl.ch), Ecole Polytechnique Federale de Lausanne (EPFL), 1015 Lausanne, Switzerland dario.floreano@epfl.ch

[1]Inertial Measurement Unit

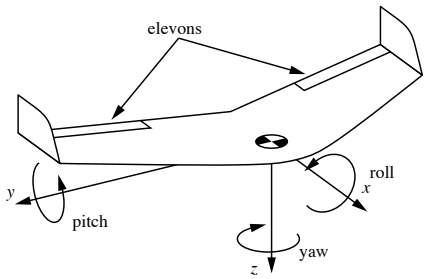

Fig. 1. Coordinate system of an aircraft reference frame with the names of the three rotation directions. In standard airplanes, the roll is controlled using ailerons on the main wing and the pitch using an elevator on the tail. Here a tail-less aircraft (also called flying-wing) is shown as it corresponds to the platform that is used for our experiments. In the case of a flying-wing, roll and pitch rotations are controlled by the differential and, respectively, common mode of actuation of the two control surfaces called elevons. These two modes of actuation are functionally identical to the ailerons and elevator of a standard airplane. The yaw is assumed to be stabilized by the vertical surface(s) or rudder(s) in order to produce so-called coordinated turns [14].

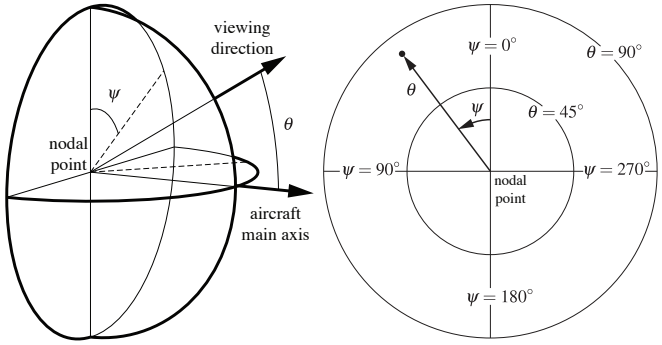

Fig. 2. The polar coordinate system that is used throughout this paper to define the viewing directions of the vision system. The main axis starts at the nodal point of the visual system (whatever is is) and is parallel to the aircraft roll axis. ψ is the azimuth angle with $\psi = 0$ corresponding to the dorsal part of the aircraft. θ is the polar angle.

aircraft can be taken into account to steer it on a predefined path while preserving the collision avoidance behaviour.

A. Optic-flow-based collision avoidance

The fundamental property of optic flow that enables proximity estimation is often referred to as *motion parallax* [16]. Essentially, it states that the component of optic flow that is induced by translatory motion is proportional to the magnitude of this motion and inversely proportional to the distance to static obstacles in the environment. It is also proportional to the sine of the angle between the translation vector and the considered viewing direction. This can be written as:

$$p_{\mathbf{T}}(\theta,\psi) = \frac{|\mathbf{T}|}{D(\theta,\psi)}\sin(\alpha), \tag{1}$$

where $p_{\mathbf{T}}(\theta,\psi)$ is the amplitude of translational optic flow measured in direction (θ,ψ) (see Fig. 2, for the polar coordinate system convention), $\mathbf{T}$ is the translation vector, $D(\theta,\psi)$ is the distance to the object seen in direction (θ,ψ) and α is the angle between the translation vector $\mathbf{T}$ and the viewing direction (θ,ψ).

The optic flow perceived by a free-flying aircraft also contains a component induced by its rotations in addition to the translational optic flow described above. Consequently, it is necessary to subtract the optic flow component due to rotations to estimate the proximity of obstacles, a process known as *derotation* of optic flow [17], [18], [19]. This can be achieved by predicting the optic flow due to rotations, as measured by rate gyroscopes, and then subtracting it from the total measured optic flow.

In standard, non-acrobatic flight, the translation vector remains essentially aligned with the aircraft main axis. The angle α in (1) can therefore be assumed to be equal to the polar angle θ (also known as *eccentricity*) of the coordinate system introduced in Fig. 2. Equation (1) can therefore be rewritten as:

$$p_{\mathbf{T}}(\theta,\psi) = \frac{|\mathbf{T}|}{D(\theta,\psi)}\sin(\theta) = \mu(\theta,\psi)\cdot|\mathbf{T}|\cdot\sin(\theta), \tag{2}$$

where μ is the proximity of the object (i.e. the inverse of distance) seen in the considered direction. Therefore, if we assume a certain flight velocity, the magnitude of translational optic flow measurements can be directly interpreted as proximity signals, scaled by the sine of the eccentricity at which the measurements are taken.

Let us now consider the directions where optic flow should be measured and how to combine these measurements to generate control signals for the aircraft. It turns out that not all the viewing directions in the visual field have the same relevance for flight control. Directions pointing at $\theta > 90°$ correspond to obstacles that are behind the aircraft and thus do not require avoidance actions. For θ values close to 0° (i.e. close to the center of the visual field), the magnitude of the optic flow measurements tends to zero because of the $\sin(\theta)$ factor in (2). These two limits ($\theta < 90°$ and $\theta > 0°$) suggest that the area of interest lies around $\hat{\theta} = 45°$. Experimental and theoretical work has shown that 45° is indeed optimal in a variety of situations [20], [18], [13]. In most practical cases, objects will not appear above the aircraft (except if it is flying down a tunnel or inverted over ground), which allows to further reduce the number of required viewing direction. We therefore consider N viewing directions on either side and below the aircraft with a certain inter-azimuthal angle $\hat{\psi}$.

In order to map optic flow estimates into control signals, *optiPilot* relies, similarly to the tangential cells of flying insects [21], [22], on a weighted sum, which can be written as:

$$c^j = \frac{\xi^j}{N\cdot\sin(\hat{\theta})}\cdot\sum_{k=0}^{N-1} p_{\mathbf{T}}(\hat{\theta}, k\cdot\hat{\psi}+\frac{\pi}{2})\cdot w_k^j, \tag{3}$$

where c^j is the j^{th} control signal (either roll or pitch control), w_k^j the associated set of weights and ξ^j a gain to adjust the amplitude of the control signal. In order to use this approach to steer an aircraft, two sets of weights $\{w_k^R\}$ and $\{w_k^P\}$, $k = 0,1,...,N-1$ must be devised, for the roll and the pitch control, respectively. Let us first consider the pitch control signal c^P (Fig. 3 top). Proximity signals from the ventral region (i.e., ψ near 180°) correspond to obstacles beneath the aircraft. The corresponding weights should thus be positive

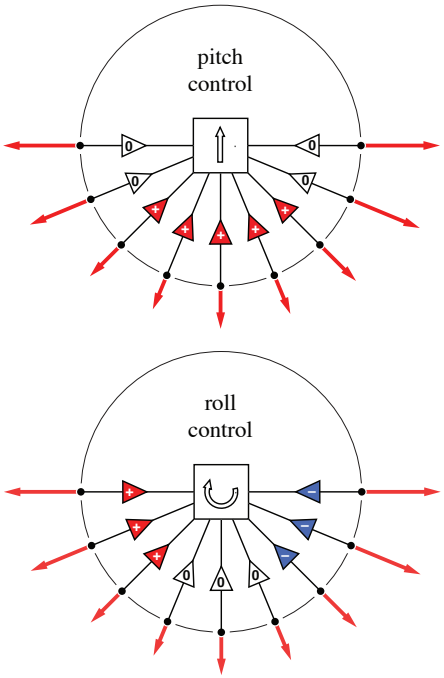

Fig. 3. Mapping translation-induced optic flow into roll and pitch control signals. The top (resp. bottom) outline represents a possible weight distribution that will make an aircraft pitch (resp. roll) away from any seen objects. The arrows in the center indicate pitch (resp. roll) direction for a positive pitch (resp. roll) signal.

to generate a positive control signal that results in a pitch-up manoeuvre. Conversely, proximity signals from either side of the aircraft (i.e., ψ near 90° and 270°) should not influence the pitching behaviour and the corresponding weights should thus be set to zero. Using a similar reasoning, one can derive the qualitative distribution needed for the weights related to the roll signal (Fig. 3 bottom). Weights corresponding to the left of the aircraft should be positive, in order to initiate a rightward turn in reaction to the detection of an obstacle on the left. Inversely, weights on the right should be negative. Since obstacles in the ventral region ($\psi = 180°$) are avoided by pitching only, the weights in this region should be set to zero. For more details on this control strategy, possible variations of the weight distributions, viewing directions and the impact of the wind, the reader is invited to consult either [13] or [19].

B. Following GPS-defined paths

The first step towards GPS-defined path following is to devise an algorithm that, based on the current position and heading of the aircraft, provides a certain turning rate and direction that steers the aircraft back on track. Nelson et al. [23] proposes a the use of a vector field that for each position in space tells the aircraft how to alter its course to get back on track. The error between the current heading of the aircraft and the vector of the field can then be used as input for a proportional controller that defines the roll angle of the aircraft and therefore its turning rate [14].

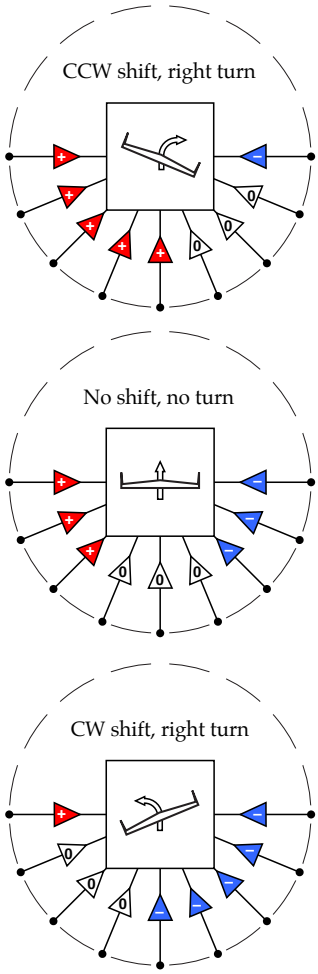

Fig. 4. Conceptual representation of steering by shifting the roll weight distribution around the roll axis. Assuming level flight over flat terrain, the effect on the roll angle will be proportional to the shift, which will lead to a subsequent steering in the same direction, according to the basics of flight dynamics [14].

As *optiPilot* does not explicitly measures the attitude of the aircraft, but rather continuously reacts to the proximity of objects, the GPS direction is taken into account by shifting the roll weight distribution $\{w_k^R\}$ around the roll axis (Fig. 4). Assuming a flat terrain underneath the aircraft, shifting this weight distribution clockwise (respectively counterclockwise) of a certain angle will result in a left (resp. right) banked attitude of the same angle, which will lead to a left (resp. right) turn. More details on this procedure can be found in [19]. As long as the shift angle remains small (i.e. <45°), the basic avoidance capabilities are maintained, while this mechanism provide a clear tendency to steer towards the desired GPS-defined path, as will be demonstrated in the following experiment.

III. Experiment Setup

A. Flying platform and sensors

The test flying platform is based on a *swinglet* from *senseFly*[2] (Fig. 5 top). The aircraft is equipped with an electronic

[2]http://www.sensefly.com

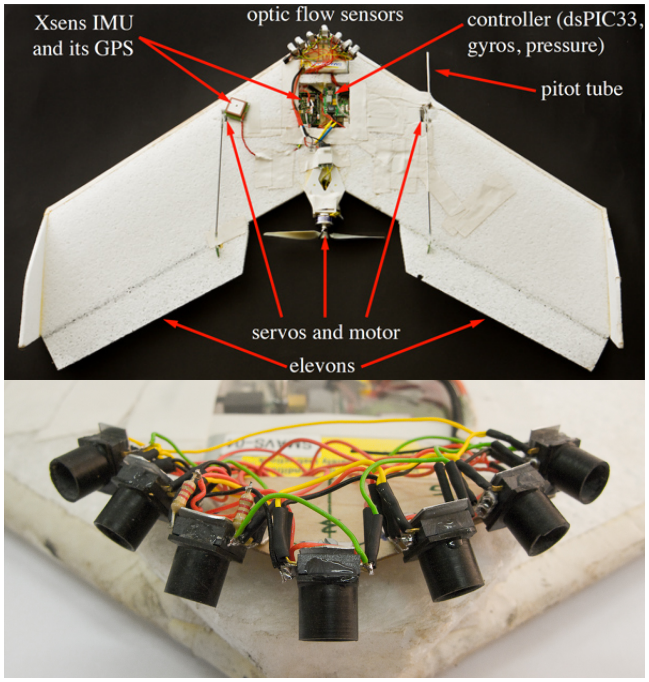

Fig. 5. Top: The *swinglet* flying wing used for the experiments. It has a wing span of 80 cm and a total weight of 407 g including about 50 g of additional sensor payload. No particular efforts have been made at this stage to reduce the weight of the sensor. Bottom: Visual front-end composed of seven optic computer mouse sensors pointed at 45° eccentricity with respect to the aircraft roll axis.

board including a Microchip dsPIC33 microcontroller, on which our control strategy is implemented. This controller is interfaced to three orthogonally-arranged Analog Devices ADXRS610 rate gyroscopes that are used for optic flow derotation (see previous section). In order to measure airspeed, it is also equipped with a Freescale MPXV5004DP differential pressure sensor and a custom-built Pitot tube. In order to regulate airspeed, a PID controller is employed to command the thrust so to maintain a cruise airspeed of approx. 12 m/s.

In order to record the state of the aircraft during the experiments, we mounted an Xsens MTi-G unit which provides a 6-degree-of-freedom state estimation. However, this unit is never used for the control of the aircraft. The states of the aircraft and its sensors are monitored and recorded in real-time using a 2.4 GHz Digi XBee-PRO radio-link, the *Ishtar* communication protocol [24] and the *e-mo-tion* monitoring software[3].

In order to measure optic flow without requiring heavy cameras and image processors, we opted for computer mouse sensors (Fig. 5 bottom), which natively output two signals that are proportional to the optic flow in their x and y axes. Seven Avago ADNS5050 sensors are mounted on a rig and pointed at 45° eccentricity (see previous section) with respect to the aircraft main axis in order to cover both the lateral and ventral regions. Note that no particular efforts have been made to ensure a precise alignment of the optic flow sensors with the theoretical viewing directions. We estimate their alignment to be within $\pm 5°$.

[3]http://gna.org/projects/e-mo-tion

To convert the reading of the mouse sensor into signals that can be fed to *optiPilot*, we first convert the image shift provided by each sensor into an optic flow vector tangent to an hypothetical unit sphere (Fig. 2), at the location towards which the mouse chip is oriented. The vector of the expected rotation-induced optic flow at this location is computed based on the measurements by made by the rate gyros. The subtraction of both vectors lead to an estimate of the direction and amplitude of the translation-induced optic flow in this particular viewing direction. The amplitude of this estimate is then fed to *optiPilot*.

B. Parameter tuning

Parameters such as the roll weight distribution, the pitch weight distribution, the pitch gain, the pitch bias (which prevents the tendency to fly upwards), the roll gain, and the steering controller gain have been empirically tuned in order to get a stable flight behaviour. Most of the values have been kept identical to the ones used in the reactive collision avoidance experiments [13]. The pitch weight distribution required some specific attention in order not to loose too much altitude during steeply-banked turns. This could be obtained by increasing the weights linked to the ventrolateral optic-flow sensors and acting on the pitch rate. The resulting distribution is as follows: $\{w_k^P\} = \{0.3; 1.8; 0.8; 0.3; 0.8; 1.8; 0.3\}$.

C. Test environment

In order to carry out our first tests, we chose a natural environment featuring a reasonably flat terrain and a few copses of trees. The background image of Fig. 6 has been taken with one of our flying platforms just after the flight tests described below. This image shows the kinds of textures that were present during the flight tests (corn fields, dirt fields, some grass, and some small concrete roads) and the two copses that we used as obstacles. During the flight tests the air was calm (wind less than 1 m/s) and the sky was partially cloudy.

IV. Results

Two flights of 25 minutes each were performed in fully autonomous mode, from take-off to landing[4], where the aircraft covered more than 32 km in 46 laps. During these two flights, the aircraft consistently flew at approx. 9 m above the ground and managed to avoid more than 90 potential collisions with the two copses of trees that were present on the GPS-defined path (Fig. 6). As can be seen in Fig. 6, the trajectory of the aircraft is remarkably repetitive (± 1.5 m) among the 23 laps of one flight. The maximum lateral error with respect to the optimum GPS path is 6 m. This shows that, although the GPS-based steering of the aircraft is done indirectly through the shifting of the weight distribution of *optiPilot*, the precision of the path following mechanism is kept at a reasonable level when flying over flat terrain. This precision also explains why the tree avoidance always occurs

[4]More details on how take-off and landing are controlled using *optiPilot* can be found in [25].

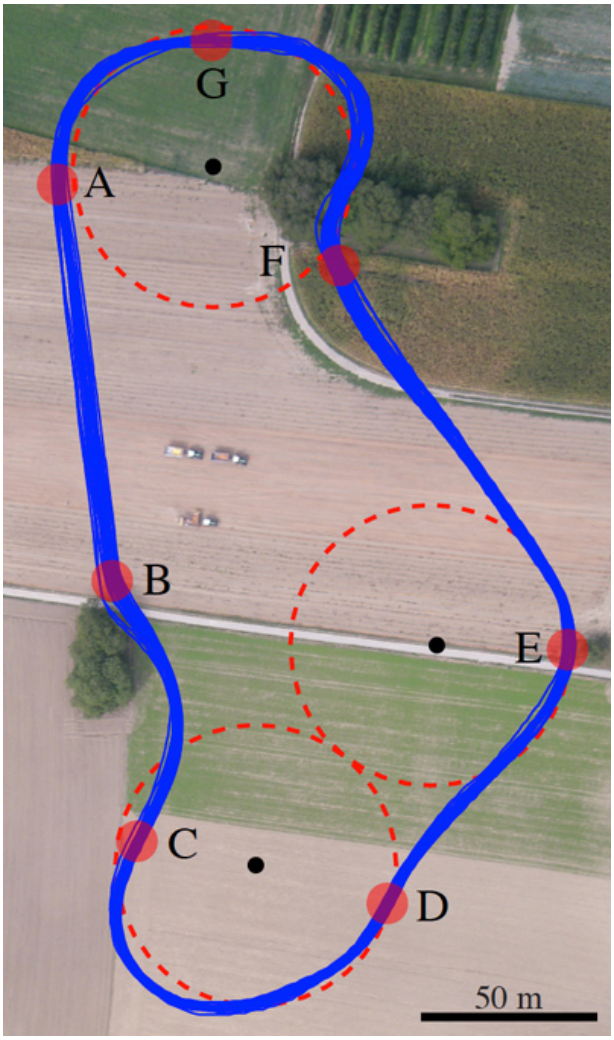

Fig. 6. Trajectory of the aircraft as recorded by the on-board GPS during a 25-min fully autonomous ride. The trajectory is overlaid on top of an aerial image of the environment taken at the same period as the experiments to show the variety of fields that were flown over as well as the two copses of trees approximately 15 m tall, which required swift avoidance actions. The dashed lines represents the circular way-points that attract the aircraft through the vector field [23]. The letters indicate a series of points of interest along the path: A is the starting point; B is the place of the first copse; C corresponds to the point where the aircraft recovers from the avoidance action caused by the trees and the start of a 180° curve that ends up in D; E is at the middle of a slight heading change; F indicates the start of the second avoidance action, which finishes in G in the middle of a turn.

in the same manner, i.e. avoiding the copse by the left in B and flying over the canopy in F. Since the aircraft is always confronted to the obstacles in the same location, arriving at the same speed and same altitude, *optiPilot* will steer the aircraft in a very similar manner around them.

Fig. 7 shows in more details what happens along a full lap in terms of optic flow, weight distribution shift, aircraft attitude and altitude. When flying straight from A to B (see the points define both on the trajectory graph of Fig. 6 and in Fig. 7), the optic flow levels tend to be high in the ventral part of the aircraft and almost null in the lateral regions. Any discrepancies from this repartition will be corrected for by the roll control mechanism of *optiPilot*. The relatively constant altitude is a result of the balance occurring between the negative pitch bias and the pitch-up induced by *optiPilot* due to the presence of ventral optic flow.

When the aircraft encounters the trees in B, the optic flow distribution shows a clear increment in the ventral-right part, which results in an immediate pitch-up (corresponding to a positive pitch angle) and roll-left (corresponding to a negative roll angle) action, which can be seen in the third and fourth row of Fig. 7. This avoidance action in fact steers the aircraft away from its GPS-defined path, which in turn leads to a non-zero kernel shift (see second row) that will attempt to counter this deviation. This counter-action, whose resulting roll angle can be seen in the third row just after $t = 30$ sec, can take place as soon as the trees are cleared.

Between C and D, one can observe a significant kernel shift up to -40° (second row) leading to a left (negative) roll angle (third row), which results in the long turning action over the grass terrain. During this turn, the optic-flow distribution is consistently shifted towards the left. A slight altitude drop can be observed in the fifth row of Fig. 7, which can be explained by the decrease in lift due the banked attitude of the aircraft that needs to be compensated for by a greater amount of ventral optic flow, and therefore a lower altitude. Then between D and F, a similar, but shorter left turn can be observed with a slight increase of altitude towards the end, due to the presence of a corn field made of plants approx. 2 m tall.

After point F, the second tree avoidance is clearly visible with increase of the pitch angle up to 20° followed by an altitude gain of about 9 m. This climbing response is superimposed with a right-left steering due mainly to kernel shifts, but also to the presence of some higher branches generating lateral optic flow. Once the copse is cleared, a dive occurs until the ventral optic flow due to the terrain generates a counter-action.

A video recording of this flight is available as accompanying material or for download at http://lis.epfl.ch/microflyers.

V. Conclusion

This paper shows how GPS-based path following can be added on top of the previously described *optiPilot* control strategy, which relies on optic flow to ensure flight stabilization and 3D collision avoidance [13]. Fully autonomous nap-of-the-earth flight at about 9 m height and 12 m/s has been demonstrated over an agricultural terrain featuring various ground textures and two copses of trees, requiring quick and efficient avoidance actions. Two fully autonomous flights of 25-minutes each with more than 100 tree avoidance actions were performed with a 400-gram unmanned aircraft fitted with seven off-the-shelf optic-flow detectors.

Future work will consist of testing our approach in more difficult settings such as in presence of medium winds or in urban environments.

Acknowledgments

We would like to thank Adrien Briod and Alexandre Habersaat for the implementation of the vector field path following algorithm, Severin Leven for the development of the embedded electronics, and Tim Stirling for the proof-reading of the manuscript. This work has been supported by the Swiss National Science Foundation (grant 200020-116149).

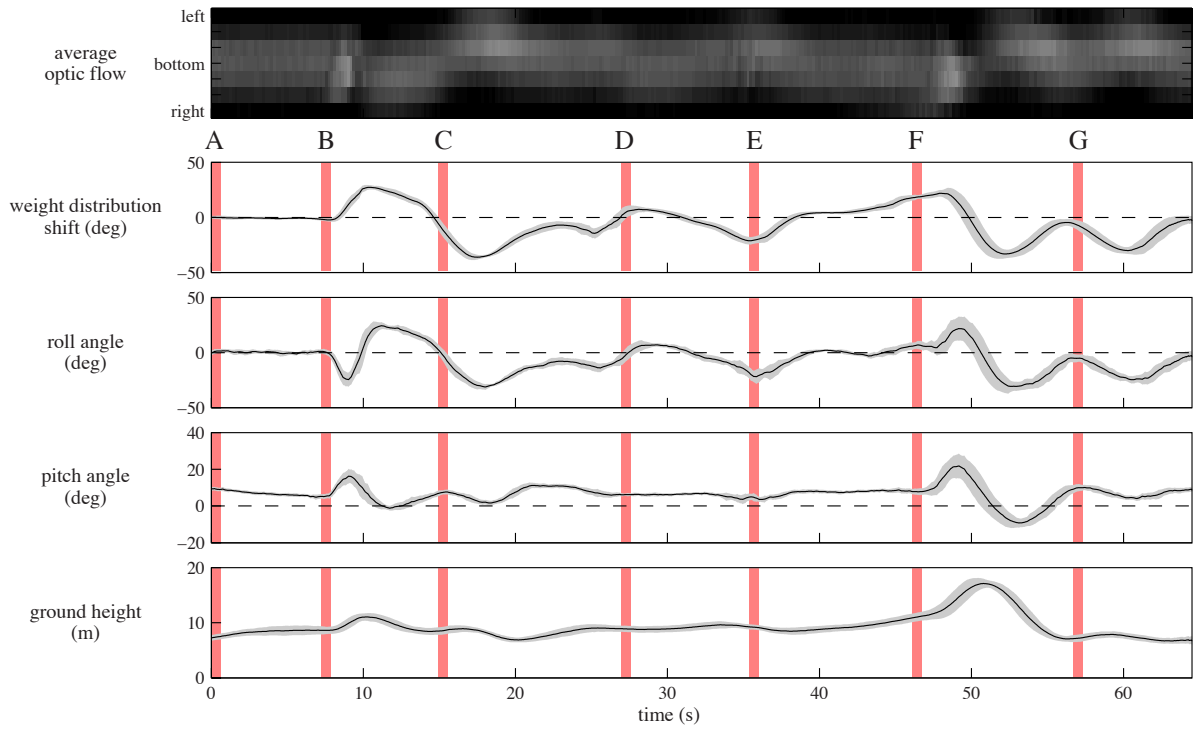

Fig. 7. Data acquired during the flight shown in Fig. 6. The first graph displays the average optic flow (in grey level, black meaning 0% and with 100%) perceived by the aircraft in each viewing directions, from left to right. The following rows show the commanded weight distribution shift, the resulting roll and pitch angles as measured by the XSens IMU and the altitude over the launching position as measured by the on-board pressure sensor. The dark lines indicate the average over 25 passages along the GPS-defined circular path. The grey zones show the standard deviation among the various passages. The letters and the vertical lines throughout the graphs correspond to the points of interest defined in Fig. 6.

REFERENCES

[1] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, and W. Johnson, "Autonomous vehicle technologies for small fixed-wing UAVs," *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92–108, 2005.

[2] L. Muratet, S. Doncieux, Y. Brière, and J. Meyer, "A contribution to vision-based autonomous helicopter flight in urban environments," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 195–209, 2005.

[3] S. Griffiths, J. Saunders, A. Curtis, T. McLain, and R. Beard, *Obstacle and Terrain Avoidance for Miniature Aerial Vehicles*, ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 2007, vol. 33, ch. I.7, pp. 213–244.

[4] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," *The International Journal of Robotics Research*, vol. 27, no. 5, pp. 549–574, 2008.

[5] R. Brooks, *Cambrian Intelligence*. The MIT Press, Cambridge, MA, 1999.

[6] T. Netter and N. Franceschini, "A robotic aircraft that follows terrain using a neuromorphic eye," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[7] 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.

[8] 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.

[9] J.-C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions on Robotics*, vol. 22, pp. 137–146, 2006.

[10] 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.

[11] A. M. Hyslop and J. Humbert, "Wide-field integration methods for autonomous navigation of 3-D environments," in *Proceedings of the 2008 AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[12] J. Zufferey, A. Beyeler, and D. Floreano, *Optic-Flow to Steer and Avoid Collisions in 3D*. Springer, 2009, ch. 6, in press.

[13] A. Beyeler, J. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous Robots*, vol. 27, no. 3, pp. 201–219, 2009.

[14] B. Stevens and F. Lewis, *Aircraft Control and Simulation*, 2nd ed. Wiley, 2003.

[15] J. Koenderink and A. van Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, pp. 247–254, 1987.

[16] T. Whiteside and G. Samuel, "Blur zone," *Nature*, vol. 225, pp. 94–95, 1970.

[17] A. Argyros, D. Tsakiris, and C. Groyer, "Biomimetic centering behavior for mobile robots with panoramic sensors," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 21–68, 2004.

[18] J.-C. Zufferey, *Bio-inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*. EPFL/CRC Press, 2008.

[19] A. Beyeler, "Vision-based control of near-obstacle flight," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2009.

[20] S. Hrabar and G. S. Sukhatme, "Optimum camera angle for optic flow-based centering response," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China*, 2006, pp. 3922–3927.

[21] H. Krapp, B. Hengstenberg, and R. Hengstenberg, "Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly," *Journal of Neurophysiology*, vol. 79, pp. 1902–1917, 1998.

[22] M. Egelhaaf and R. Kern, "Vision in flying insects," *Current Opinion in Neurobiology*, vol. 12, no. 6, pp. 699–706, 2002.

[23] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[24] A. Beyeler, S. Magnenat, and A. Habersaat, "Ishtar: a flexible and lightweight software for remote data access," in *Proceedings of the 2008 European Micro Air Vehicle Conference EMAV08*, 2008.

[25] A. Beyeler, J.-C. Zufferey, and D. Floreano, "optiPilot: control of take-off and landing using optic flow," in *Proceedings of the European Micro Air Vehicle Conference EMAV09*, vol. 27, 2009, pp. 201–219.