Toward 30-gram Autonomous Indoor Aircraft: Vision-based Obstacle Avoidance and Altitude Control*

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

***Abstract*— We aim at developing autonomous micro-flyers capable of navigating within houses or small built environments. The severe weight and energy constraints of indoor flying platforms led us to take inspiration from flying insects for the selection of sensors, signal processing, and behaviors. This paper presents the control strategies enabling obstacle avoidance and altitude control using only optic flow and gyroscopic information. For experimental convenience, the control strategies are first implemented and tested separately on a small wheeled robot featuring the same hardware as the targeted aircraft. The obstacle avoidance system is then transferred to a 30-gram aircraft, which demonstrates autonomous steering within a square textured arena.**

***Index Terms*— Indoor flying robot, optic flow, steering control, collision avoidance, altitude control.**

I. Introduction

We describe vision-based control strategies for obstacle avoidance and altitude control that have been tested separately on terrestrial and aerial robotic platforms, paving the way toward fully autonomous slow-flyers [13] capable of navigating within houses or small built environments. Flying indoor implies a number of challenges that are not found in outdoor autonomous flight. These include small size and slow speed for maneuverability, light weight to stay airborne, low-consumption electronics, and smart sensing and control to fly in textured environments. Sensors commonly employed in robotics, in particular range finders, are too heavy and energy consuming for use on very light flying platforms. Therefore, we take inspiration from flying insects for the selection of sensory modalities, and for the choice of navigational behaviors, which often co-evolved in order to simplify the processing of sensory input.

The basic requirements for an autonomous flight consist in course stabilization, obstacle avoidance, and altitude control. Flies can satisfy the first requirement using their halteres, which basically act as gyroscopic sensors, allowing them to stabilize roll, pitch and yaw movements [3], [11]. The last two requirements are solved by vision and in particular optic flow (OF) [4]. There is evidence that image expansion is used as a criterion to trigger quick turning actions, also called saccades, that serve as collision avoidance reflex [18]. It has also been shown that honeybees use ventral image velocity to control their velocity and altitude in the landing phase [17].

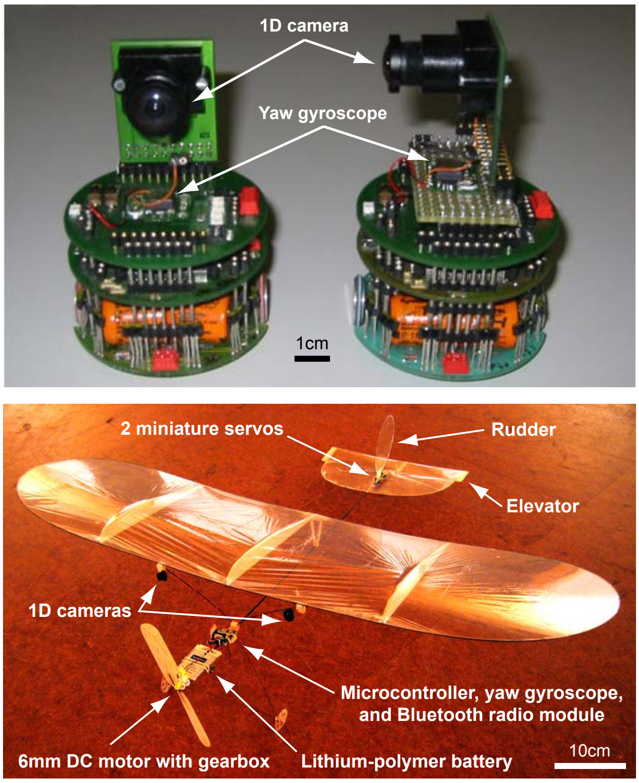

Fig. 1. The robotic platforms. *Top*. Two Khepera robots, the left one equipped for obstacle avoidance (steering) and the right one for lateral wall following (corresponding to altitude control in the case of a flying robot). Gyroscopes for yaw rotation detection are visible on the top turrets, just below the 1D cameras (one-dimensional array of pixels). *Bottom*. 30-gram indoor slow-flyer equipped with a yaw gyroscope and two horizontal 1D cameras pointing at 45° off the longitudinal axis of the aircraft. This plane flies around 2m/s and features a minimum turning radius of approximately 1.3m. The energy source is provided by a 310mAh Lithium-polymer battery. Overall power consumption is around 2W.

In this paper, we demonstrate OF-based obstacle avoidance first with a small Khepera robot (Fig. 1, top), then with a 30-gram indoor aircraft (Fig. 1, bottom). Note that the Khepera does not require any course stabilization mechanism since it is in contact with the ground, whereas the plane needs a mechanism to ensure straight flight when no obstacles are detected. So far, the altitude control algorithm has been implemented on the wheeled robot by

*This work is supported by the Swiss National Science Foundation.

transposing the problem into a wall following situation. Both robotic platforms are equipped with the same kind of electronic components so that the control mechanisms implemented on the Khepera can easily be downloaded into the plane microcontroller for further testing in flight [20].

One of the main problem in using OF for estimating distances from objects in the scene is the spurious signals introduced by rotations. For an observer in a static environment, the OF is function of the 3D movement (rotation and translation) and of the distances from surrounding objects. However, only OF due to translation (TransOF) depends on distances. The rotational optic flow (RotOF) is thus a spurious signal, which contaminates the OF field. In this paper, we highlight the interference between RotOF and TransOF and show how the problem can be solved by merging gyroscopic information with vision.

Several groups used insect visual-control systems as models for wheeled robots, either for obstacle avoidance [5], [9] or corridor following [16]. Some of these robots used active camera mechanisms for stabilising their gaze in order to minimize RotOF. All of them relied on the fact that they were in contact with a flat surface in order to infer or control their self-motion through wheel encoders. Since flying robots have no contact with ground, those approaches cannot be applied without modifications. Furthermore, the tight weight budget of indoor aircrafts precludes active camera mechanisms for gaze stabilization. Finally, all the above mentioned robots, except the machine based on analog-electronics by Franceschini and colleagues [5], featured off-board image processing and control.

A few noticeable attempts were made in using OF for aerial navigation. Specific studies on altitude control were conducted in simulation [10], in tethered helicopters [14], and with outdoor unmanned air vehicles [2], but none of those effectively tackled the problem of RotOF introduced by pitch rotations. Another work in simulation [12] demonstrated full 3D navigation of a minimalist model of an insect. Some preliminary trials of obstacle avoidance and automatic landing were carried out with a small model plane [6], but experiments were limited to very specific situations and no continuous autonomous flight was reported so far.

In the remaining of this paper, Section II presents the typical patterns of frontal and ventral OF occurring when moving toward a flat surface or flying over ground. This theoretical study is based on fictitious, large field of view (FOV), 2D cameras. In practice, such cameras would be too heavy and require too much processing power for a 30-gram flying robot. This study is thus essentially aimed at defining where are the regions of interest in the visual field and what kind of OF patterns can be used for triggering obstacle avoidance actions or maintaining safe altitude over ground. Section III describes the 1D OF detector implementation together with the underlying algorithm running in the tiny on-board microcontroller. Section IV presents the experiments and results obtained with the real robots.

II. From Optic Flow to Control Strategies

For a vision sensor moving within a stationary environment, the 2D projection of the relative 3D motion of scene points onto its retina is called motion field. This motion field depends on the motion of the vision sensor and the structure of the environment (distances from objects). In general, the apparent motion of the image, the so-called OF, does not necessarily correspond to the motion field, in particular when no contrast is detectable in a given viewing direction or because of the aperture problem [19]. However, the distinction between OF and motion field will be ignored in this preliminary theoretical analysis. In practice, we assume enough contrast on the surrounding surfaces and integrate OF signal over time such to reduce the impact of losing it during a short period of time.

The 3D motion of a rigid body can always be characterized unambiguously by a translation vector (specifying direction and velocity) and a rotation vector (describing the axis of rotation and angular rate). When the vision sensor is moving in a stationary environment, the OF vector on its imaging surface in a given viewing direction (often specified in spherical coordinates as an azimuth and elevation) depends on the distance to the object seen in that direction. The OF vector in each viewing direction can be described as a vectorial sum of two distinct components, one due to the translation, TransOF, and another due to the rotation, RotOF [7]. The first one is inversely proportional to distances from objects, while the latter does not depend on distances. It follows that if the translation is known and the rotation is null, it is possible to estimate distances in each viewing directions, and thus avoid obstacles. However, it is quite common in free-maneuvering agents that RotOF overwhelms TransOF, thus making extraction of distance information difficult. Fig. 2 qualitatively illustrates the effect of rotational movement on TransOF field generated by a forward motion toward a flat surface (expanding flow field).

It has been shown that some flies fly in straight trajectories during which they estimate OF in order to decide when to trigger a saccade [18]. In order to reduce as much as possible the disturbing effect of rotations, we rely on the same strategy to control the robots. During straight motion, altitude can be maintained by regulating ventral OF [14]. During saccade, OF is ignored since it is dramatically affected by RotOF, which can produce values beyond OF detector range. However, it is not possible

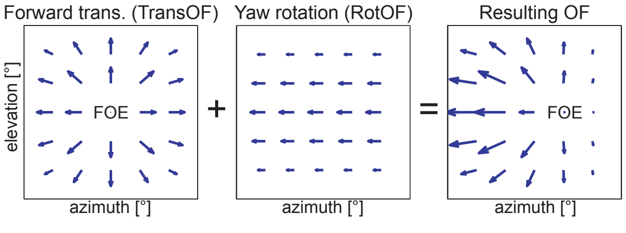

Fig. 2. An example of the effect of RotOF on TransOF. In particular, notice the shift of the focus of expansion (FOE).

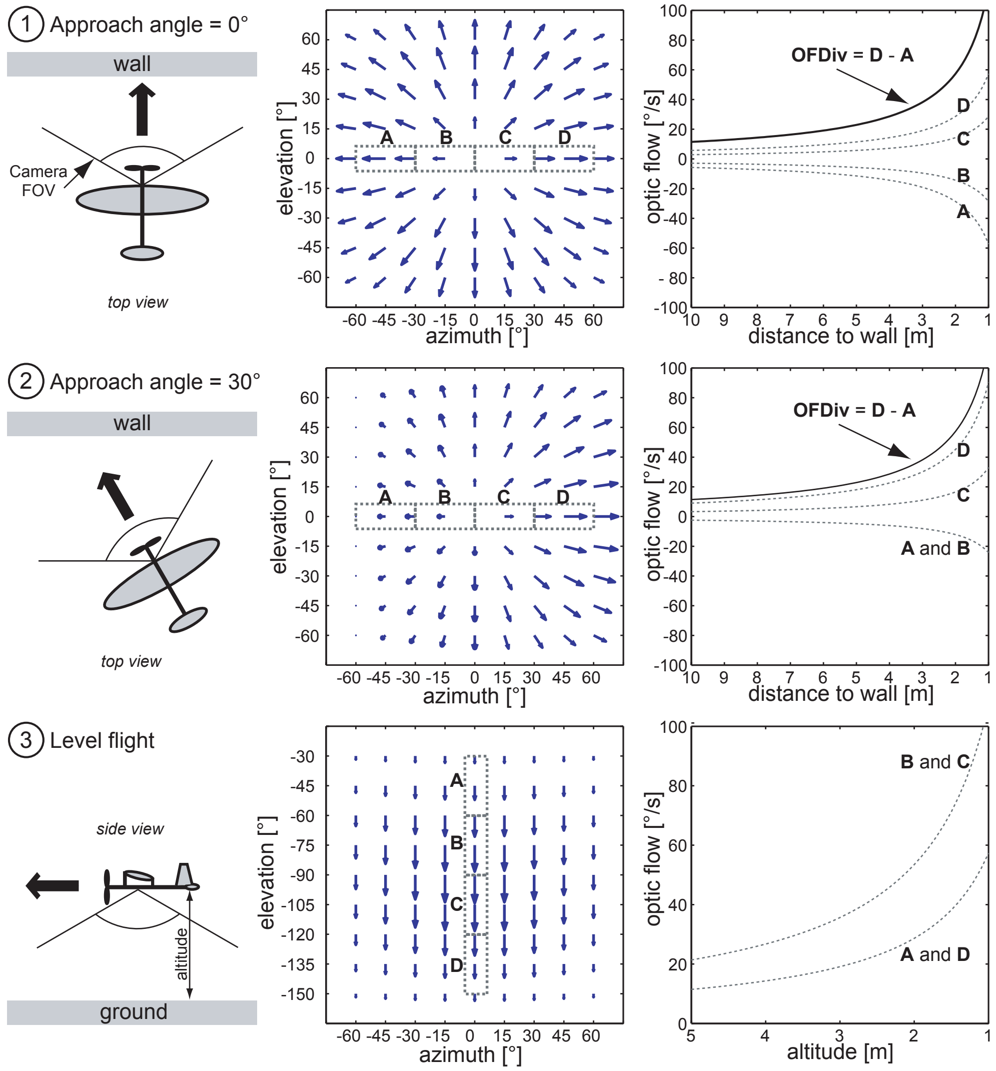

Fig. 3. Patterns of OF as viewed from fictitious 2D cameras with a 120° FOV. *Case 1.* Frontal approach toward a wall as viewed by a frontal camera. *Case 2.* Same as previous but with an oblique angle of approach of 30°. *Case 3.* Ventral camera looking downward and OF generated at different altitude. The first column depicts the camera position and orientation as well as the airplane trajectory. In these examples, the aircraft is flying at 2m/s in pure forward translation. The second column gives snapshots of the typical OF patterns occurring in each situations. Indicated with dashed rectangles are four regions of interest (compartments labelled from A to D) that could be covered by 1D OF detectors. The third column shows the OF amplitude as a function of the distance from the wall or from the ground at the center of each of the four compartments. The curves labelled OFDiv are the difference between right (D) and left-most (A) regions (see text for details).

to ensure absolutely null rotation during straight motion due to small heading corrections, pitching adjustments for altitude control, or turbulences. Since RotOF does not depend on distance, it is possible to retrieve that part of OF by knowing the rotation rate. Therefore, in our robots, RotOF is continuously deduced during straight motion using information provided by a gyroscope (see Section III).

Assuming that residual RotOF can be discounted from global OF, we can now concentrate on typical TransOF patterns in order to devise the rules allowing for obstacle avoidance and altitude control. Fig. 3 shows theoretical TransOF fields for ideal 2D cameras with a 120° FOV. Keeping in mind that 2D vision is too expensive for our indoor flying robot, which has a payload of about 8g for the electronics and sensors, dashed lines on the center graphs indicate interesting regions in the FOV that could be covered by one or several 1D sensors, which can be managed by the on-board, low-power microcontroller.

When the robot is moving toward a textured wall, the field is divergent with an amplitude inversely proportional to the distance [7], as shown in the right column of Fig. 3, where OF amplitudes are plotted for a set of four different viewing directions corresponding to the regions labelled A, B, C and D. This information can be used to trigger a turning action in order to avoid the wall. Since TransOF is relatively small around the focus of expansion (FOE) in the center of the image, we rely on more lateral regions where OF is maximal, i.e., regions A and D. By subtracting signed values of the right OF from that of the left OF, we obtain a rough estimate of the OF divergence (OFDiv), which will be used as the criterion to trigger collision avoidance. OFDiv is independent of the angle at which the plane approaches the wall, as can be seen by comparing the curves in the first and second row of Fig. 3. This way of measuring OF field divergence is reminiscent of the minimalist method proposed in [1]. Another interesting feature of the expanding flow field is that by comparing the absolute values of left and right TransOF, it is possible to determine the side experiencing higher OF, which corresponds to the side closer to obstacles (see for instance Fig. 3, second row). As in flies [18], this information is used to decide whether to turn right or left.

The third row in Fig. 3 illustrates the OF field generated in the ventral part of the FOV when the plane is flying over flat ground. All OF vectors are parallel and maximum amplitude is located in the center of the FOV where the photoreceptors are closer to the ground. Since TransOF amplitude is inversely proportional to altitude, it is possible to control altitude of the plane by maintaining a given TransOF amplitude for the ventral camera. It is worth noticing that the locus of maximum TransOF corresponding to the perpendicular distance from the ground depends on the pitch angle of the aircraft. Therefore, we take into account the region where the TransOF is maximum as representative of the actual altitude.

III. Optic Flow Detector

Rather than covering large part of the FOV with a single large FOV vision sensor (fish-eye-view with multiple glass-lenses optics or omnidirectional mirror-based system), we opted for a much simpler and lighter solution based on 1D cameras and single plastic lenses pointing at directions of interest, as identified in the previous section. 1D cameras also present the advantage of having few pixels, hence keeping the computational and memory requirements within the limits of a simple microcontroller. We call "OF detector" the system made of a 1D camera connected to the embedded microcontroller together with the processing algorithm. The front-end camera (TAOS inc. TSL3301) consists in a CMOS sensor with a 1D array of 102 grey-level pixels, out of which only 50 can be used because of optics limitation (see [20] for further details on this camera, possible optics, and comparison with other vision systems). The 8-bit embedded microcontroller is a Microchip$^{\text{TM}}$ PIC18F6720 running at 20MHz, with 3840 bytes of RAM, 64kWords of program memory, without floating-point capabilities.

In order to measure OF, we adapted the image interpolation technique proposed by Srinivasan [15], which estimates the global motion in a given region of the image by a single-stage, non-iterative process. This algorithm computes the displacement s that provides the best fit between a linear combination of two shifted versions of a reference image and a new image acquired after a small delay $\triangle T$. The ratio $\frac{s}{\triangle T}$ (in pixel/s) can be related to the actual OF (in deg/s) by knowing the corresponding inter-pixel angle (depending on the optics). If $I(n)$ denotes the grey level of the n^{th} pixel in the 1D image array, the algorithm computes the amplitude of the translation s between an image (or a subpart of it) captured at time t, $I_t(n)$, and a later image captured at time $t+1$, $I_{t+1}(n)$. It assumes that, for small displacements of the image, $I_{t+1}(n)$ can be approximated by $\hat{I}_{t+1}(n)$, which is a weighted linear combination of the image $I_t(n)$ and two shifted versions $I_t(n \pm 1)$ of the same image:

$$\hat{I}_{t+1}(n) = I_t(n) + s\frac{I_t(n-1) - I_t(n+1)}{2}. \qquad (1)$$

Note that $\frac{I_t(n-1)-I_t(n+1)}{2}$ is taken as a discrete substitute for the image gradient $\frac{dI}{dn}$. The image displacement s is then estimated by minimizing the mean square error E between the estimated image $\hat{I}_{t+1}(n)$ and the actual one $I_{t+1}(n)$, with respect to s:

$$E = \sum_n \left[I_{t+1}(n) - \hat{I}_{t+1}(n)\right]^2, \text{ and } \frac{dE}{ds} = 0 \Leftrightarrow \qquad (2)$$

$$s = 2\frac{\sum_n \left[I_{t+1}(n) - I_t(n)\right]\left[I_t(n-1) - I_t(n+1)\right]}{\sum_n \left[I_t(n-1) - I_t(n+1)\right]^2}. \qquad (3)$$

In our case, the delay $\triangle T$ between t and $t+1$ is such to ensure that the actual shift does not exceed ±1 pixel. $I_t(n \pm 1)$ are simply generated by translating the reference image by one pixel respectively to the left and to the right.

Equation (3) is implemented in the microcontroller, which grabs two successive images corresponding to $I_t(n)$ and $I_{t+1}(n)$ with a delay $\triangle T$ of a few milliseconds (typically 5 to 10ms). This process is repeated for each sensory-motor cycle, whose length depends on the application (typically between 50 and 100ms). Every pixel intensity is encoded on 8-bit and variables used in (3) are 32-bit integers. The image processing is accomplished in fixed-point and lasts only 1.6ms for 50 pixels. Of course, the number of pixels (and the corresponding image region) on which the algorithm is applied can be freely chosen.

In order to adjust the delay $\triangle T$, we compared the output of the gyroscope (see Fig. 1, AnalogDevices$^{\text{TM}}$ ADXRS150, measuring up to 150°/s angular rate) with OF values in pure yawing rotations [21]. After calibration, the plane equipped with its two horizontal OF detectors (left and right) was brought into the test arena (Fig. 6) and randomly rotated by hand around its yaw axis, i.e., without translational movement. Fig. 4 shows the sensory outputs during this experiment and demonstrates the almost perfect match between gyroscope and RotOF. This result legitimates the cancellation of RotOF by simply subtracting gyroscopic value from OF detector outputs.

IV. Experiments and Results

A. Obstacle Avoidance on Wheels

In the first experiment, the obstacle avoidance strategy is tested with the Khepera (Fig. 1, top) in a square environment (Fig. 5). The camera is equipped with a fish-eye lens providing an horizontal 1D FOV of 120°, which

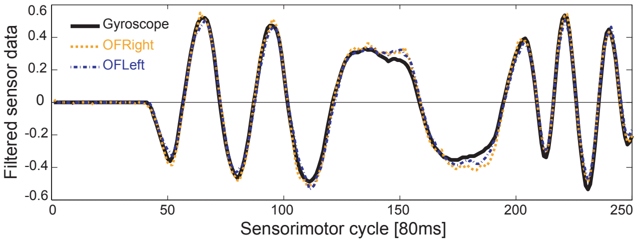

Fig. 4. Graph showing the good correlation between gyroscopic values and OF estimates. The data have been recorded during about 20sec (every 80ms) from the two OF detectors (left and right) mounted on the plane (see Fig. 1, bottom). The robot was held by hand in the experimental arena (Fig. 6) and randomly rotated around its yaw axis. A value of 0.5 on the y-axis corresponds to 100°/sec.

is divided into four subparts corresponding to the regions shown in Fig. 3. Only the two most lateral parts of the FOV are used for obstacle avoidance. OF estimation algorithm is applied independently on these two regions corresponding to 12 pixels each. OFDiv signal (introduced in Section II) is computed by subtracting OF from the right part (labelled D in Fig. 3) to the one from the left part (labelled A).

The control algorithm is composed of two behaviors: (i) straight forward motion at constant speed (10cm/s) during which the system continuously estimates OFDiv, (ii) rotation for a predetermined amount of time (1sec) without taking care of any sensory information. The transition from state (i) to state (ii) is triggered whenever OFDiv reaches a threshold, whose value has been determined experimentally. As proposed in Section II, the rotation direction of turning actions is determined by the asymmetry between left and right OF, i.e., the Khepera turns away from the side experiencing highest OF value.

The robot was able to navigate without collisions for more than 45 minutes (60'000 sensory-motor cycles), during which 84% of the time was engaged in straight motion and the remaining 16% in turning actions. Fig. 5 shows a typical trajectory of the robot during this experiment.

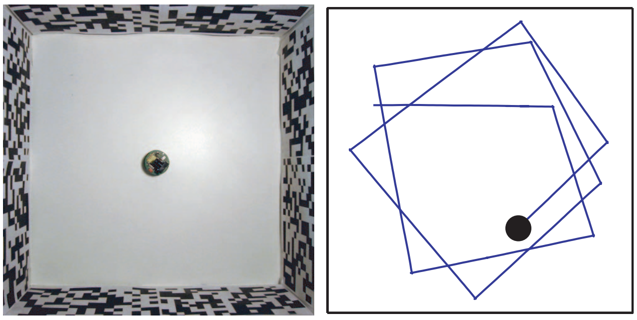

Fig. 5. Obstacle avoidance with the Khepera equipped with frontal camera. *Left*. 60 by 60cm arena used for obstacle avoidance experiments. The walls are covered with randomly generated, high contrasted patterns. *Right*. Path of the robot in autonomous steering mode: straight motion when no obstacles are detected and saccadic turning action when image expansion (OFDiv) reaches a fixed threshold. The black circle represents the Khepera in its starting position. The path has been reconstructed from wheels encoders.

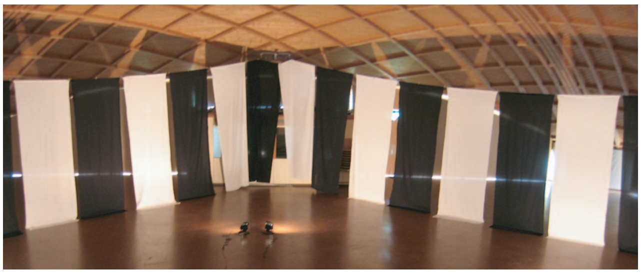

Fig. 6. Part of the square arena (16 by 16 meters) used for obstacle avoidance experiments with the flying robot. The walls are made of black and white curtains. Note that the regularity of the pattern is due to the default material size.

B. Obstacle Avoidance in the Air

In the second experiment, the algorithm is assessed on the 30-gram indoor aircraft (Fig. 1, bottom) flying in the square arena illustrated in Fig. 6. Instead of using only one frontal camera with an heavy fish-eye lens (5g), the plane is equipped with two horizontal 1D cameras (1g each) pointing at 45° off the longitudinal axis. Only 28 pixels are used in each camera, for a corresponding FOV of 2x 40°.

The aerial robot is expected to fly along straight trajectories interspersed with predefined turning actions, using the same control algorithm described for the Khepera. Concerning state (i), the only difference is that the plane must actively drive its rudder in order to fly straight (course stabilization). This is achieved by means of a proportional, negative feedback loop connecting the yaw gyroscope to the rudder servomotor. It is worth noticing that, ensuring absolutely null rotation during straight motion is not possible. However, this is not critical since RotOF is continuously deduced from OF estimates using information coming from the gyroscope. The second state, i.e., the saccade itself, is programmed as a series of motor commands for a predefined amount of time. During 1 second, the motor is set to full power, the rudder deflection follows an experimentally optimized curve up to full deflection, and the elevator is slightly pulled such to compensate for the decrease in lift due to the roll angle.

So far, the 30-gram robot has been able to fly collision-free for more than 4 minutes without any intervention (except for maintaining altitude, which was done manually using a joystick and the Bluetooth wireless communication). Only 20% of the time was engaged in turning actions, which indicates that the plane flew always in straight trajectories except when very close to the walls. During those 4 minutes, the aerial robot generated 50 saccades, and covered about 300m in straight motion. Video clips are available in the video proceedings or on the web page *http://phd.zuff.info*. More details about experiments with the plane will be published elsewhere [22].

C. Altitude Control as Wall Following on Wheels

In order to assess the altitude control mechanism suggested in Section II, we implemented it as a wall following

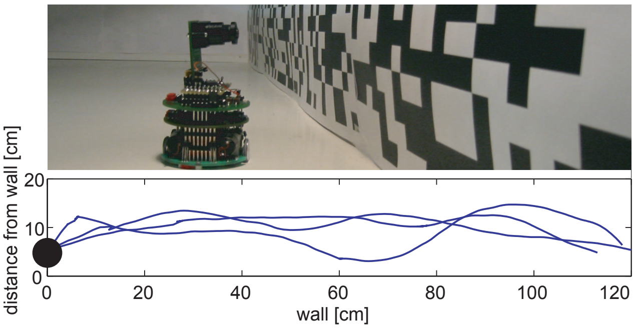

Fig. 7. Altitude control (implemented as wall following on a wheeled robot) with the Khepera equipped with lateral camera. *Top.* 120cm long setup for wall following experiments (Khepera with lateral camera). *Bottom.* Wall following results (3 trials). The black circle indicates the robot initial position. Trajectories are reconstructed from wheel encoders.

mechanism in the Khepera with a lateral camera and fish-eye optics (Fig. 1, top-right). In this case, the distance from the wall would correspond to the altitude of the aircraft. A proportional-derivative controller tries to maintain a constant TransOF value by acting on the heading of the robot, which would correspond to the pitch angle of the airplane. The measured TransOF feeding the controller is always the highest among the four part of the FOV in order to take into account only the actual perpendicular distance from the flat surface, as described in Section II.

Several runs have been carried out on a 120cm-long wall (Fig. 7). Although the robot does not always maintain the same distance from the wall, this preliminary experiment shows that a very simple controller can produce reliable ground avoidance.

V. Conclusion and Outlook

We presented a bio-inspired approach to vision-based obstacle avoidance and altitude control. Although the altitude control has not yet been integrated on the 30-gram airplane, the experiments carried out so far indicate that the proposed strategy is well adapted to navigation with very limited processing power (the microcontroller is about 4 orders of magnitude slower than a Pentium IV) and under drastic weight constraints. We are currently integrating the altitude control system onto the flying platform in order to demonstrate fully autonomous flight.

One of the main problems we are currently facing is the dependence on high-contrast textured environments. To address this issue, we established collaboration with the Institute of Neuroinformatics in Zurich aimed at developing analog VLSI vision sensors [8] with adaptive photoreceptors and built-in OF detection. Those will feature quick adaptation to ambient light while being lighter and far less power demanding than the current 1D cameras, allowing for mounting more of them on the plane in order to widen the part of the FOV covered by the compound eye of our artificial flying insect.

Acknowledgment

The authors wish to thank C. Mattiussi and A. Beyeler for useful discussions and comments on the manuscript, and J.-D. Nicoud for his support in building the 30-gram airplane.

References

[1] N. Ancona and T. Poggio. Optical flow from 1d correlation: Application to a simple time-to-crash detector. In *Proceedings of Fourth International Conference on Computer Vision*, pages 209–214, 1993.

[2] J.S. Chahl, M.V. Srinivasan, and H. Zhang. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *The International Journal of Robotics Research*, 23(2):101–1102, 2004.

[3] M.H. Dickinson. Haltere-mediated equilibrium reflexes of the fruit fly, drosophila melanogaster. *Philosophical Transactions: Biological Sciences*, 354(1385):903–916, 1999.

[4] M. Egelhaaf and A. Borst. A look into the cockpit of the fly: Visual orientation, algorithms, and identified neurons. *The Journal of Neuroscience*, 13(11):4563–4574, 1993.

[5] N. Franceschini, J.M. Pichon, and C. Blanes. From insect vision to robot vision. *Philosophical Transactions of the Royal Society B*, 337:283–294, 1992.

[6] W.E. Green, P.Y. Oh, and G.L. Barrows. Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2004.

[7] J.J. Koenderink and A.J. van Doorn. Facts on optic flow. *Biological Cybernetics*, 56:247–254, 1987.

[8] J. Kramer, R. Sarpeshkar, and C. Koch. An analog vlsi velocity sensor. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 413–416, 1995.

[9] M.A. Lewis. Visual navigation in a robot using zig-zag behavior. In *Neural Information Processing Systems 10*. MIT Press, 1998.

[10] F. Mura and N. Franceschini. Visual control of altitude and speed in a flying agent. In *From Animals to Animats III*, pages 91–99. MIT Press, 1994.

[11] 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*, 175:695–708, 1994.

[12] T.R. Neumann and H.H. Bülthoff. Behavior-oriented vision for biomimetic flight control. In *Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics*, pages 196–203, 2002.

[13] J.D. Nicoud and J.C. Zufferey. Toward indoor flying robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 787–792, 2002.

[14] F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans*, pages 2339–2346. T. J. Tarn and T. Fukuda and K. Valavanis, April 2004.

[15] M.V. Srinivasan. An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71:401–416, 1994.

[16] M.V. Srinivasan, J.S. Chahl, K. Weber, S. Venkatesh, and H. Zhang. Robot navigation inspired by principles of insect vision. In A. Zelinsky, editor, *Field and Service Robotics*, pages 12–16. Springer-Verlag, 1998.

[17] M.V. Srinivasan, S.W. Zhang, and J.S. Chahl. Landing strategies in honeybees, and possible applications to autonomous airborne vehicles. *The Biological Bulletin*, 200:216–221, 2001.

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

[19] A. Verri and T. Poggio. Against quantitative optical flow. In *Proceedings of the First International Conf. on Computer Vision (ICCV)*. Computer Society Press, Washington DC, 1987.

[20] 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*, 2003.

[21] J.C. Zufferey and D. Floreano. Optic-flow-based steering and altitude control for ultra-light indoor aircraft. Technical report, 2004.

[22] J.C. Zufferey and D. Floreano. Fly-inspired visual steering of an ultra-light indoor aircraft. *IEEE Transactions on Robotics*, submitted.