

Design, Aerodynamics, and Autonomy of the DelFly

G.C.H.E. de Croon², M.A. Groen¹, C. De Wagter¹,
B. Remes¹, R. Ruijsink¹, and B.W. van Oudheusden^{1*}

Abstract

One of the major challenges in robotics is to develop a fly-like robot that can autonomously fly around in unknown environments. In this article, we discuss the current state of the DelFly project, in which we follow a top-down approach to ever smaller and more autonomous ornithopters. The presented findings concerning the design, aerodynamics, and autonomy of the DelFly illustrate some of the properties of the top-down approach, which allows the identification and resolution of issues that also play a role at smaller scales. A parametric variation of the wing stiffener layout produced a 5% more power-efficient wing. An experimental aerodynamic investigation revealed that this could be associated with an improved stiffness of the wing, while further providing evidence of the vortex development during the flap cycle. The presented experiments resulted in an improvement in the generated lift, allowing the inclusion of a yaw rate gyro, pressure sensor, and microcontroller onboard the DelFly. The autonomy of the DelFly is expanded by achieving (1) an improved turning logic to obtain better vision-based obstacle avoidance performance in environments with varying texture, and (2) successful onboard height control based on the pressure sensor.

^{*}(1). Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HT Delft, The Netherlands, (2). Advanced Concepts Team, European Space Agency. Email: guido.de.croon@gmail.com

1 Introduction

There is a considerable research effort to develop fly-like robots with the ability to autonomously fly around in unknown environments. The challenge derives from the fact that flying locomotion requires the robot to continuously react to its environment in real time, while the light weight and small size of the robot significantly limits the energy, sensors, and processing onboard. In addition, flapping wing propulsion is a fascinating research area in itself, where notwithstanding the extensive attention it has received over the years, many aspects still merit further attention.

Essentially, there are two main approaches to creating fly-like robots: bottom-up and top-down. In the bottom-up approach [47, 48, 54] one starts by creating all the tiny parts that are deemed important to a fly-sized ornithopter. The most remarkable example of this approach is the 60 mg robotic insect developed at Harvard University, which can produce sufficient thrust to take off vertically [54]. This is achieved by using Smart Composite Microstructures (SCM). The robotic insect was still fixed to taut guide wires that restricted the robot to vertical motion, and it was still powered externally. In future work, the group plans to allow all degrees of freedom and to incorporate onboard energy supply, sensors, and processing.

In the top-down approach, one starts with a fully functioning (relatively large-scale) ornithopter [4, 18]. By studying this ornithopter, theoretical insights can be gained into the necessary properties for a smaller version. Research then progresses by creating and analyzing ever smaller systems, while always maintaining a fully functioning flying robot. One advantage of this approach is that it allows interplay between theory and practice. Especially in the field of artificial intelligence, having a physical and fully functioning robot is of great value [13, 41, 40]: real-world tests force the experimenters to take into account all aspects of the robotic system. In addition, they reveal physical properties of the system that can be exploited by the algorithms.

In this article, we discuss the current state of our research on the DelFly, which is an insect-flight inspired Micro Air Vehicle (MAV) developed at the Delft University of Technology (cf. [18]). Figure 1 shows an image of the ‘DelFly II’ in flight. The inset shows the different onboard electronic components. The DelFly is part of our top-down approach in which we integrate the insights of empirical tests in the areas of aerodynamics and autonomous flight. In Section 2, we introduce a novel design of the wing structure, which

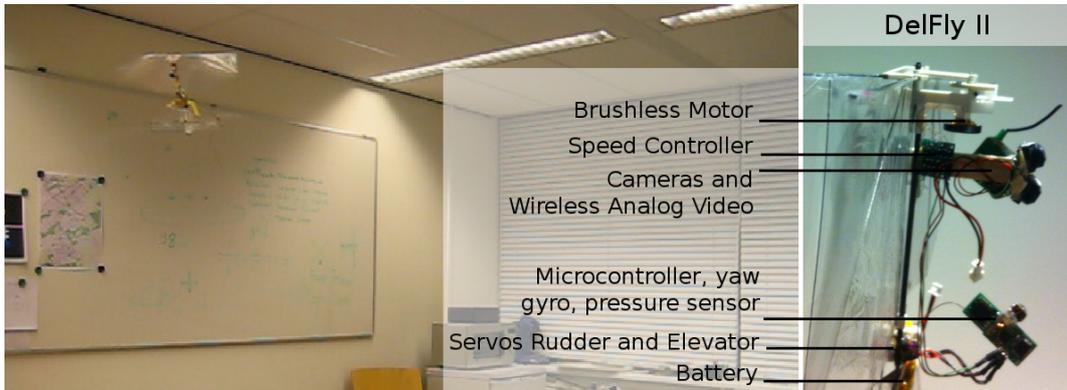


Figure 1: Photo of the 16-gram, 28 cm wing span DelFly II in flight. The inset shows the electronic components onboard.

improves the thrust-to-power ratio of the DelFly. In Section 3 we discuss the main aerodynamical findings that led to the current design of the DelFly, and focus on the air flow around the new wing structure. In Section 4, we explain our approach to achieving autonomous flight on the DelFly, focusing on the particular challenges posed by a flapping-wing MAV. Experiments are performed in which the DelFly (1) uses an improved turning logic that is suitable for environments with sparse texture, and (2) uses an onboard pressure sensor for achieving height control. Conclusions are drawn in Section 5.

2 Design

The design of the DelFly is inspired by flapping-wing insect flight. Flapping wings simultaneously generate lift and thrust, and entail a favourable manoeuvrability and large flight envelope [23, 21]. Indeed, the flight envelope of the DelFly II ranges from forward flight at 7 m/s to hover flight and even backward flight at 1 m/s. A key feature of the DelFly design is its biplane-wing configuration, with two pairs of wings placed above each other (see Figure 2). The wings flap in anti-phase, which provides stability advantages compared to a single pair of wings. However, the wing configuration also largely determines the aerodynamic behaviour and is crucial for the flight performance. A better understanding of the aerodynamic characteristics will allow us to further decrease the overall size of DelFly, while preserving its excellent flight performance.



Figure 2: Photo of biplane-wing configuration of the DelFly II.

The specific configuration under study here is the ‘DelFly II’ (Figure 1 and 2), which has a mass of 16 grams (including power pack and onboard camera) and a wing span of 28 cm. Each wing is constructed from a relatively stiff carbon-rod leading edge spar and a thin Mylar foil with two additional stiffeners. The leading-edge rods are driven actively, while the wing foil rotates during flapping due to the passive deformation under combined inertial and aerodynamic loads.

Most early design choices have been made on the basis of empirical research, measuring the generated lift and thrust-to-power ratios for different parameter choices. For example, the optimal flap angle and frequency for hovering have been determined experimentally for the DelFly II [12]. The most power efficient operation occurs at a maximum flap angle of 30° to 36° , however in its present configuration the DelFly has a maximum flap angle of 44° to maximize the payload lifting capability. Some of the experiments have shown that seemingly minor design decisions can have a large impact on the flight performance. The best example in this respect concerns the rod that forms the wing’s leading edge. The rod’s shape has an enormous impact on the produced lift forces, determining whether or not the DelFly is able to perform the hover flight mode. A ‘D’-shape with the round side facing ‘forward’ was found to lead to the best performance. The so-constructed leading edge essentially acts as a spring tensioning the foil, providing more stiffness in the flapping direction than in the wing direction.

In this section, we discuss the optimization of the wing geometry to obtain

a better thrust-to-power ratio on the DelFly II. The comparison between the old and the new wing of the power and thrust measurements are discussed in Subsection 2.1. A preliminary analysis of the in-flight wing shape is presented in Subsection 2.2.

2.1 Wing geometry study

The space of all possible wing geometries is vast. As a consequence, any empirical study of wing geometry will have to limit the considered parameters. In the presented wing geometry study, parameters such as wing area and wing shape were kept constant. Instead, the study focused on the locations and orientations of the stiffeners in the wing, as this was expected to have a considerable influence on the thrust-to-power ratio of the design [28]. For the force and power measurements, a custom made microcontroller board has been used. The microcontroller board was used to guarantee a high and constant sampling frequency and was connected to a PC with a serial connection. Zemic load cells connected to the microcontroller board were used to measure the forces on the DelFly model with a measurement resolution of 1.92 mN. The standard deviation of the force measurements was of the order of 1.5 mN (which is of the order of 1% of the thrust force to sustain hovering flight).

As measure of performance the ratio of thrust over power consumption was used. The improved wing resulting from the wing geometry study (see Figure 3) showed a 5% improvement in thrust-to-power ratio with respect to the original DelFly wing. The difference between the original and improved wing is especially found in the more favourable power consumption. The thrust generation of both wings is approximately equal (at the same flapping frequency) but the improved wing has a 5% reduction in power consumption.

The influence of stiffener thickness was also investigated. Increasing the stiffener thickness resulted in a higher thrust generation, but a lower thrust-to-power ratio. Also fixing the stiffeners to the leading edge led to a higher thrust, but lower thrust-to-power ratio, which is therefore thought to be an effect of the increased stiffness. Changing the area distribution to a more bat-like appearance, where the location of the maximum chord is changed from the root to a more outboard position, had a positive effect on the thrust-to-power ratio, but the wing did not reach the thrust level needed for sustaining hovering flight.

To further improve the DelFly's flight performance also a new driving

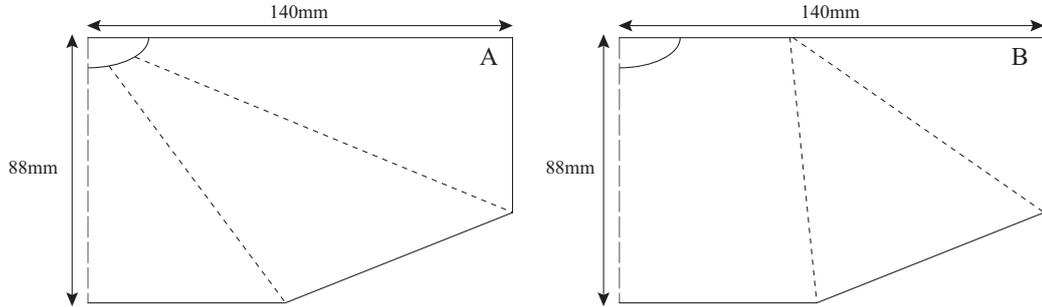


Figure 3: Stiffener location and orientation of the original wing (A) is changed for the improved wing (B), while the wing surface area and layout is kept unchanged

mechanism was developed. The new driving mechanism is made from polycarbonate, and manufactured using injection molding. This mechanically made driving mechanism is much more robust, easier to reproduce, 35% lighter, and is made with smaller tolerances than the original handmade mechanism. Since the motor and gears operate in the same plane as the hinges, the new driving mechanism is also more efficient.

The improvements in power consumption are summarized in Figure 4 (for these measurements the thrust generation is approximately equal at the same flapping frequency). For the DelFly model operating at a flapping frequency of 13 Hz (hovering conditions), the improved mechanism resulted in a 20% power reduction and the improved wing for a 5% power reduction. As a consequence, in total an improvement of 33% in the thrust-to-power ratio was obtained.

The aerodynamic power is computed as the total power consumption in air minus the power consumption in vacuum [12]. With the improved mechanism, the aerodynamic power is $76\% \pm 2\%$ of the total power consumption in air; this value is the same for both wings and approximately constant along the operational frequency range. At a flapping frequency of 13 Hz, the remaining 24% of the power consumption consists of 16% for the power required to drive the motor, gears and hinges and 8% for the power required to overcome the elastic-inertial forces. These numbers indicate a “dynamic efficiency” of $\frac{(100-24)}{(100-16)} = \frac{76}{84} = 90\%$, which agrees favorably with the predictions from the model of Weis-Fogh (1973) [36], considering that for the DelFly both aerodynamic and inertial bending moments are of comparable order.

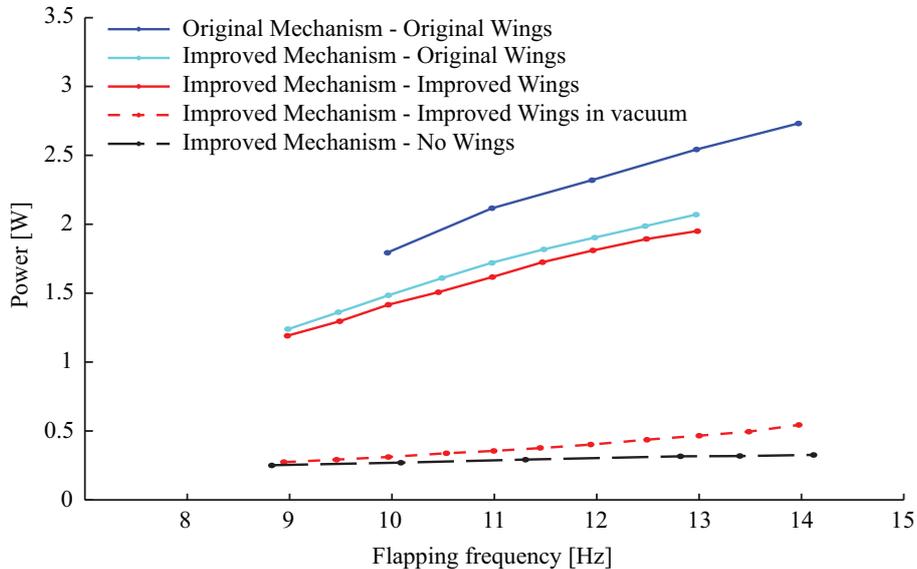


Figure 4: Average power consumption, showing the improvements from the new wings and new driving mechanism (the standard deviation on these results is negligible and hence not shown).

2.2 In-flight wing shape

The DelFly wings are made from Mylar foil with carbon stiffeners and a D-shaped carbon rod for the leading edge. The in-flight wing shape is determined by aerodynamic, elastic and inertial forces. The aerodynamic forces are in turn influenced by the wing shape, leading to a complex fluid-structure interaction. Determination of the in-flight wing shape is important to help explain aerodynamic effects.

The air flow around the wings is investigated with Particle Image Velocimetry (PIV), which allows the in-flight wing shape to be extracted from the images taken with the PIV cameras of the aerodynamic study (see Section 3). Using the PIV set-up at low laser intensity without seeding, a cross-section of the wing is illuminated. In Figure 5 the wing shape of the original DelFly wing at various moments during the flap cycle is shown as function of the non-dimensional time, $\tau = t/T$, where T is the flapping period. The cross-sections are taken at spanwise location, $b = 0.71R$, with R the wing semi-span, and at a wing flapping frequency of 11 Hz. The cross-sections in Figure 5 show the foil folded over the D-shaped leading edge carbon rod

(mentioned in the introduction of this section). The orientation of the carbon rod gives rigidity in the stroke direction but allows the leading edge to bend up and down (in chordwise direction) more easily. This enables the wing to perform a heaving motion during flapping, shown by the leading edge path in Figure 5.

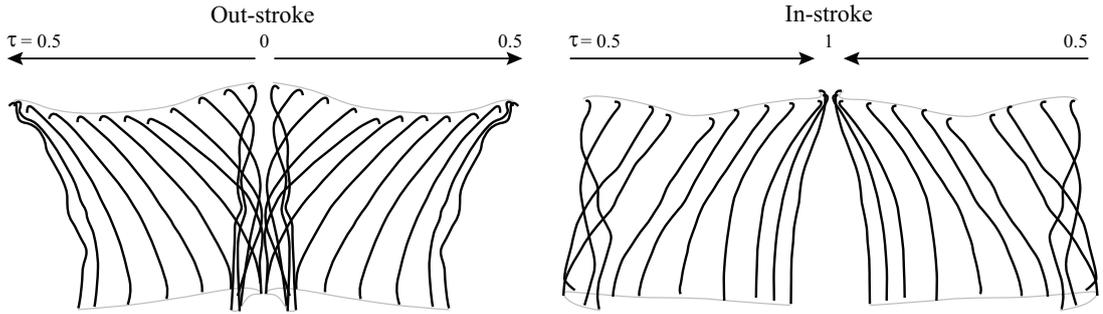


Figure 5: Cross-sections of the original wing during a flap cycle at a flapping frequency of 11 Hz at a spanwise location of $0.71R$. Loaded with an average thrust of 0.15 N

Another feature shown in Figure 5 is the clap-and-peel between the two wing pairs. During the out-stroke, from $\tau = 0$ to $\tau = 0.30$, while the leading edges move apart, the upper and lower wing surfaces peel apart and at the trailing edge the wing foil claps together. Since the leading edges are the part of the wing being driven by the flapping mechanism, the wing motion can be seen as a forced displacement of the leading edge where the rest of the wing is being dragged behind, like a flag being waved, hence, illustrating the large impact of the wing flexibility.

While comparing the in-flight wing shape of the original wing with that of the improved wing, see Figure 6, it can be seen that the original DelFly wing is more flexible during the rotation ($\tau = 0.5$ to 0.6) than the improved wing. The stiffeners on the improved wing are placed more outward compared to the stiffeners on the original wing. This gives the improved wing more rigidity at locations near the wing tips. The more flexible behaviour of the wing during rotation could also be due to a difference in foil tension, caused by differences in the mounting of the wings to the DelFly body. Foil tension is difficult to control accurately because of submillimeter tolerances and the strong aging effect. The wing shape of both wings during the translation phases (both in-stroke as out-stroke) is comparable.

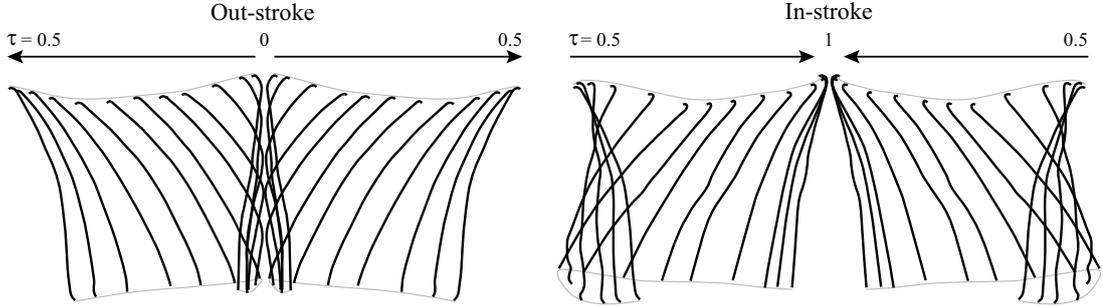


Figure 6: Cross-sections of the improved wing during a flap cycle at a flapping frequency of 11 Hz at a spanwise location of $0.71R$. Loaded with an average thrust of 0.14 N

3 Aerodynamics

In this section, we present the results of our study of the air flow around the new wing structure in the hover condition. For this experimental investigation, force and flow field measurements are performed on a hovering DelFly II. The flow field is studied using PIV. The aerodynamic mechanism generating forces on the DelFly are related to those found in insect flight. Since leading edge vortices (LEVs) in insect flight are identified as the most important unsteady aerodynamic mechanism enhancing lift generation for insects [23], the aerodynamic investigation predominantly focuses on vortex development around the DelFly. Although the DelFly operates in Reynolds number regime ($O(10^4)$ in hovering flight) which is quite large with respect to average insect species, it is expected that the unsteady flow features induced by the flapping-wing motion are driven by vortical dynamics that may be quite comparable to those in insect flight. A second important aerodynamic mechanism increasing thrust generation for the DelFly, is the clap-and-peel mechanism. This lift-enhancing wing-wing interaction, also found on various insects [33], was first described by Weis-Fogh in 1973 [36].

3.1 Leading edge vortices

Besides conventional steady means of lift generation, insects use additional unsteady aerodynamic mechanism to enhance the lift [33]. Due to the small thickness of the airfoil, the flow can separate directly from the leading edge

for high angle of attacks. Instead of stalling completely, the stall is delayed during the flapping motion and the flow reattaches further down the airfoil, to form a leading edge vortex (LEV). The LEV forms a suction force on the upper surface that increases both lift and drag [44], see Figure 7.

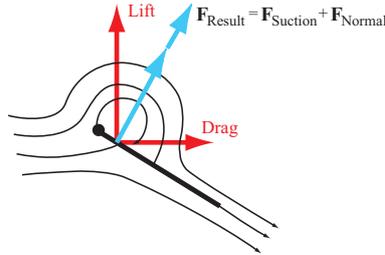


Figure 7: Leading edge suction. The leading edge vortex on a thin airfoil gives an extra suction force parallel to the normal force giving rise to extra lift and drag. Adapted from Sane [44]

Extensive research has been done to investigate the LEV in the flight of the hawkmoth, *Manduca Sexta* [53, 50]. It was found that the LEV started from a condition of dynamic stall and formed a conical leading edge spiral vortex. An explanation for the stability of the LEV is that a spiralling axial flow within the vortex core transports energy into the tip vortex. Later research by Birch et al. [9] on a robotic wing showed the LEV flow structure and spanwise flow to depend on Reynolds number. Where a spanwise flow at $Re = 1400$ was present, at $Re = 120$ it was not observed. The LEV on various insects may therefore appear as (or be composed of) several flow structures.

3.1.1 Vortex development on the DelFly

The flow field around the DelFly wings is studied using stereoscopic PIV, which provides all three velocity components in the plane of the laser sheet. The in-plane velocity components are used to investigate the vortex dynamics in the cross-sectional plane normal to the wing leading edge. The out-of-plane velocity component represents the velocity component parallel to the leading edge. The PIV measurements were performed in a phase-locked manner, with acquisition triggered by the DelFly motor controller, and covering a flap cycle by in total 34 different phases. Three flapping frequencies were investigated (9, 11 and 13 Hz) and five spanwise locations of the imaging

plane (between 42 and 100% of the wing semi-span). More details on the experimental setup can be found in [28].

For the investigation of the vortex development, the vortices need to be identified and quantified. Vortex strength can be quantified by calculation of the vorticity (curl of the velocity vector field). The vorticity however, does not only identify vortex cores but also shearing motion within the flow. The location of vortex cores and vortex strength are therefore determined from the swirling strength, which is calculated according to the method of R.J. Adrian et al. [2]. The swirling strength of a local swirling motion is quantified by λ_{ci} , the positive imaginary part of the eigenvalue of the local velocity gradient tensor.

The vortex development is studied for various wings, at various flapping frequencies and at various spanwise positions. In Figure 8 the swirling strength at various moments during the flap cycle is shown, for the improved wing flapping at 13 Hz at spanwise position $0.71R$.

Figure 8 illustrates the development of the leading edge vortex (LEV) and the trailing edge vortex (TEV). It can be seen that halfway during the out-stroke an LEV is generated (C). This vortex grows larger along the chord towards the trailing edge and another LEV starts to grow from the leading edge (D). The latter LEV grows while the vorticity of the first LEV is dissipated (E). At the end of the out-stroke (F) when the wing decelerates for rotation the LEV decreases in strength. At the beginning of the in-stroke (G) the LEV from the out-stroke appears to be dissipated. Also approximately halfway during the in-stroke an LEV is generated (I). Again the LEV grows larger and is shed towards the trailing edge, while a new LEV starts to grow (J). This LEV grows (K) until the leading edges touch (L). At the beginning of the out-stroke the vortices move above the leading edge (A), interact and dissipate when the leading edges start moving apart again (B).

The LEV development seems to approximately follow the same pattern for both in-stroke and out-stroke. The LEV during the out-stroke appears more close to the wing surface. This could be due to the downward velocity generated by the peel, which lowers the angle of attack. Furthermore it can be seen that while the LEVs from the out-stroke remain approximately at the same place and are dissipated during rotation, the LEVs from the in-stroke travel above the wing where they interact with each other.

During both in-stroke and out-stroke also a trailing edge vortex (or starting vortex) is generated. These strong TEVs dissipate more slowly and are slowly shed into the wake. During the out-stroke the development of the

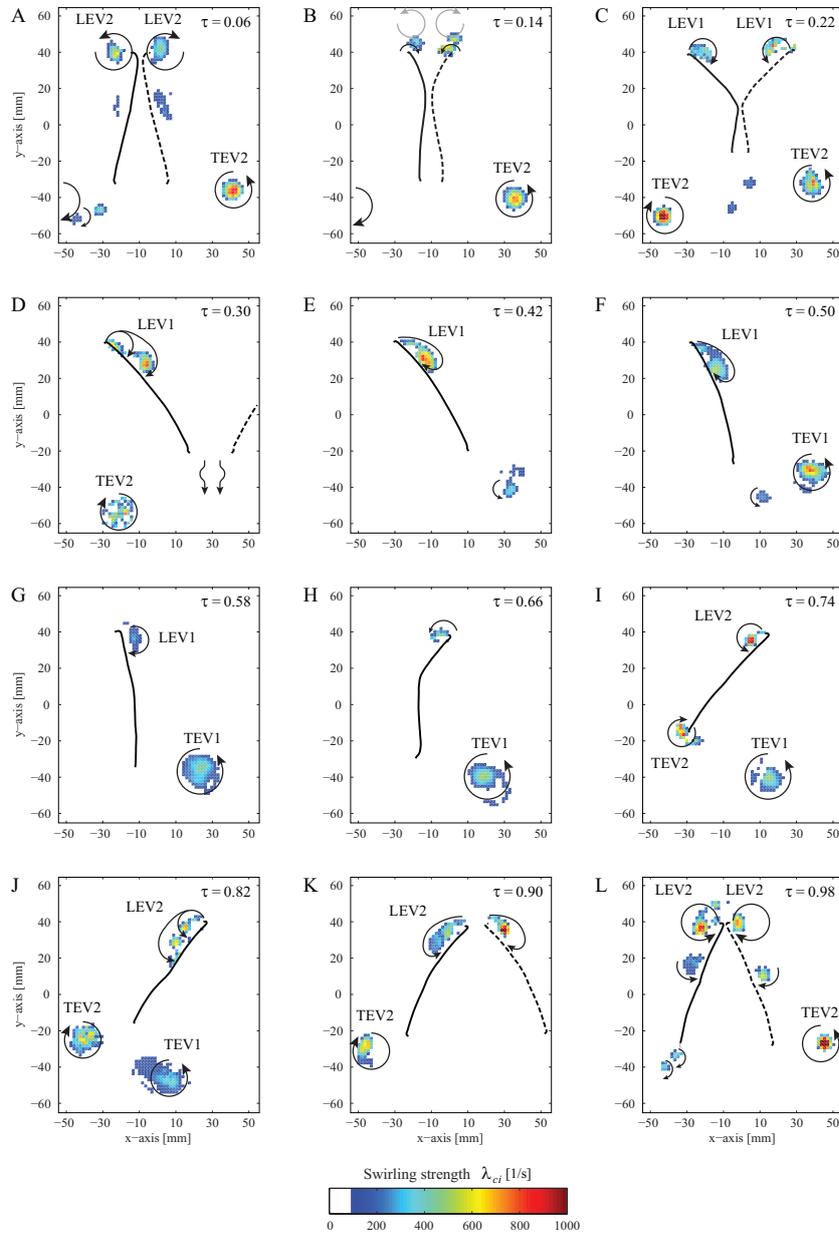


Figure 8: Swirling strength at various moments during the flap cycle for the improved wing flapping at 13 Hz and at spanwise location $0.71R$. Showing leading edge vortices (LEV) and trailing edge vortices (TEV), generated during the out-stroke (1) and in-stroke (2). Swirling direction as indicated in the figure is extracted from the velocity vector field images

TEV is postponed due to the clap-and-peel wing interaction phase. When the trailing edges separate the TEV appears to start from a complex fluid structure (D). This vortex appears to grow larger by merging with shed vorticity from the trailing edge as long as the wing is translating (E-F). During the in-stroke the TEV does start at the trailing edge (I) and grows larger and is shed from the trailing edge during translation (J).

The LEV development does not appear to be completely consistent with that described for insect flight [23, 9]. This could be due to the relative high Reynolds number. DelFly operates at a Reynolds number of 15,000 while insects fly at Reynolds numbers varying from 10 to 10,000. The higher Reynolds number could be the cause that the initial vortex is given enough time to grow relatively large, partly shed and another LEV is grown.

When considering the vortex development along the wing span, it was found that the LEV for the DelFly wing develops conically along the leading edge. The LEV is first visible at outward positions, where the translational velocity is higher, and at a later stage of the flap cycle at more in-board locations. Since the vortices at outward positions start at an earlier stage, they grow larger and are also shed at an earlier stage, where it is interesting to note that the TEV is completely shed into the wake at more in-board positions whereas it is able to follow the trailing edge more outward. The vortex tube does not extend all the way to the wing tip. While the LEV is still clearly visible at $0.86R$, it has disappeared at the wing tip. Here the LEV vortex tube has probably become connected to the tip vortex and bent towards the trailing edge. Inspection of the spanwise velocity component shows a spanwise flow directed outward inside the vortex tube. The magnitude of the spanwise flow is approximately the same as that of the maximum translational velocity at that spanwise location. A schematic representation of the results of the spanwise vortex development and spanwise flow measurements is shown in Figure 9.

3.2 Clap-and-peel

Another important aerodynamic mechanism for the DelFly (also influencing vortex development) is the clap-and-peel mechanism. The clap-and-peel mechanism is a variation on the clap-and-fling mechanism found on various insects that maximize their wing stroke by allowing physical contact between their wings at the end of the up-stroke [33]. For the DelFly this mechanism is present between the upper and lower wings during the end of the in-stroke

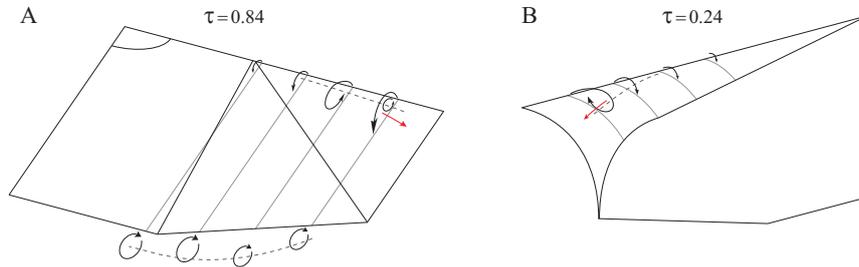


Figure 9: Sketch of the spanwise vortex development during in-stroke (A) and out-stroke (B) for the improved wing flapping at 13 Hz, where the dashed line is an indication of the vortex tube and the red arrow an indication of the spanwise flow in the vortex tube

and begin of the out-stroke, see Figure 6.

A schematic representation of the clap-and-fling is shown in Figure 10. At the end of the up-stroke (B) the leading edges of the wings touch each other before they pronate. The wings rotate around their leading edges until the wings are parallel (C), where air is expelled down from the closing gap to form a momentum jet enhancing lift. When the gap between the wings is closed the circulation of both wings cancel each other out. This ensures that the trailing edge vorticity is greatly attenuated or even absent. Since the trailing edge vorticity that is shed as a starting vortex would slow down the build-up of circulation during the next stroke (a phenomenon known as the Wagner effect), lift can now build up more rapidly [44].

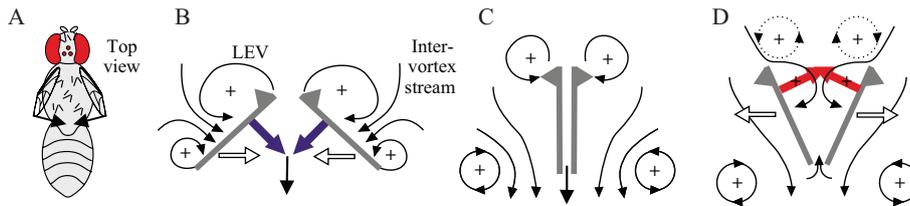


Figure 10: Schematic representation of the clap-and-fling mechanism. At the end of the up-stroke (B) the wings clap together (C) and fling apart (D). Adapted from Lehmann [34]

During the second part of the motion the wings pronate around their trailing edges and the leading edges fling apart (D). This creates a low pres-

sure region between the wings, which causes air to be sucked in and which is thought to give an initial impulse to the start of the build-up of new circulation, generating a stronger LEV. This circulation is opposite on both wings, so Kelvin’s law is satisfied and in this phase there is no need for a starting vortex [44].

For the DelFly the fling phase is different from that in Figure 10. Flexibility of the DelFly wings makes the wings peel apart under influence of a strong fluid-structure interaction. This kinematic pattern has been termed clap-and-peel. During the peel, the elastic wings physically touch, closing the gap between the wings and preventing fluid from being sucked upward. In this sense it could be said that the actual clap, where the trailing edges clap together, is postponed such that the clap and the peel take place simultaneously.

Thrust measurements performed on the DelFly showed a higher translational peak during the out-stroke, which is probably due to this clap-and-peel. As a result the measurements showed an average increase in thrust of 8% for wings that use clap-and-peel, with respect to isolated wings.

The flow field measurements showed that the peeling of the wings creates a down flow as well as a spanwise flow in-board. The clap of the wings creates a downward momentum jet, which is also thought to increase thrust generation. The flow field is shown in Figure 11 where the vectors represent the local velocity direction and magnitude and the background colour indicates the absolute velocity (absolute length the of the in-plane velocity components). Masks are applied at regions where the PIV image was obscured by reflections or the image of the wings, which makes the velocity measurement unreliable.

Vortex development is also affected by the clap-and-peel as was already seen in Figure 8. The LEV appears closer to the wing surface due to the strong down flow, reducing the effective angle of attack. At certain moments during the out-stroke, this completely prevents flow separation at the outward spanwise position. The TEV development is postponed during the start of the out-stroke as long as the clap of the trailing edges is not completed.

4 Autonomy of the DelFly

The design and aerodynamics of flapping wing MAVs have a large impact on the algorithms necessary for achieving autonomous flight. Most often,

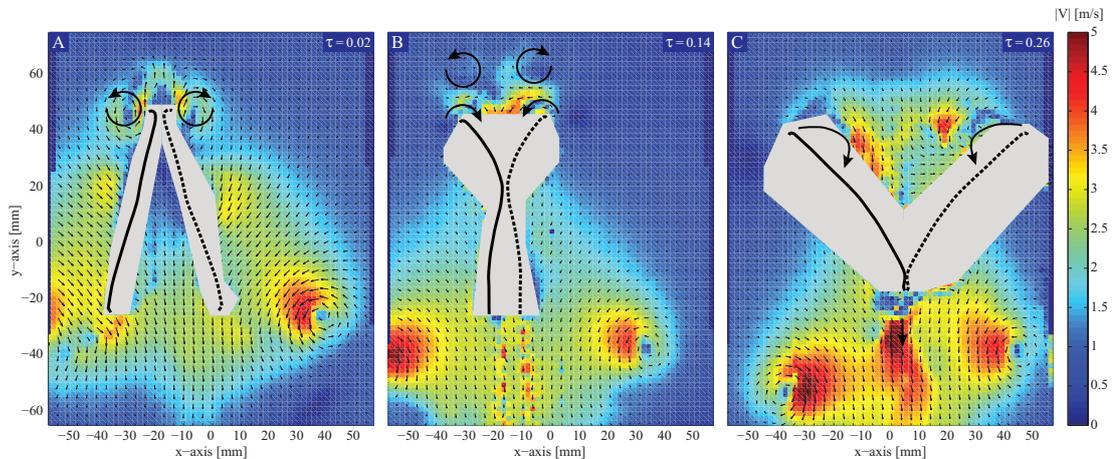


Figure 11: Velocity vector field at three moments during the clap-and-peel for the improved wing flapping at 13 Hz and at spanwise location $0.71R$

MAV autonomy is approached from an engineering point of view. Position, attitude, and velocity are measured directly and algorithms try to measure the 3D position of obstacles in the environment. When MAVs get smaller, this approach becomes increasingly difficult because of the lack of resources. Due to the light weight of flapping wing MAVs (16 grams for the DelFly II), one cannot adopt a typical Simultaneous Localization And Mapping (SLAM) approach that relies on a miniature laser range finder to achieve both obstacle avoidance and navigation [5, 29]. A possible avenue to scale down the engineering approach is to further develop visual SLAM methods, in which the state of the MAV (3D position and attitude) is estimated by ‘matching’ camera images to known locations in a 3D-model of the environment [17, 3, 15, 10]. However, the algorithms still have problems with drift [10] and are computationally expensive. This latter disadvantage may prevent the approach from being suitable for the envisaged fly-sized flapping wing MAVs in the future.

For light-weight MAVs a bio-inspired approach becomes increasingly interesting. Such an approach does not require the 3D-mapping of the environment, but relies on light-weight, energy efficient sensors and a simple behavioral repertoire to still achieve various (rudimentary) tasks. Typically the MAV directly responds to incoming visual inputs [8, 35], which is computationally very efficient. Generally, optic flow is used [31, 43, 30], since it has

been shown to play an important role in insect flight [25, 16]. For example, it is well-known that many animals rely on optic flow for obstacle detection and avoidance [27, 52, 42, 46]. In particular, the flow of image points away from the Focus of Expansion (FoE) can be used for estimating the time-to-impact to an obstacle (cf. [49]).

Currently, algorithms using optic flow for obstacle avoidance have the following two limitations. The first limitation is that the successful determination of optic flow requires sufficient texture in the environment [55]. Both the detection of obstacles and the turning logic of the algorithms depend on the presence of texture. Outdoor environments mostly have sufficient texture, but indoor environments often lack the necessary texture. Indeed, experiments with optic flow are typically held in specially prepared textured spaces [55, 43, 30].

The second limitation is that time-to-impact estimates rely on accurate optic flow measurements. Unfortunately, images made with flapping wing MAVs are challenging (cf. [7]). Currently, a small camera onboard the DeFly is used that transmits its images analogically to a ground control station. Consequently, the images contain various types of noise. Besides thermal and other measurement noise, the images also undergo noise from interfering transmission sources such as WiFi networks. However, the most difficult properties of the images derive from the flapping-wing movements. Despite the biplane wing configuration, there is residual motion up, down, and rotationally (due to slight asymmetries in the construction). This motion interacts with the line-by-line-recording of the CMOS camera¹, constantly leading to considerable image distortions. Example images are shown in Figure 12. These images are particularly affected by the flapping movements: the (straight) edge of the closet in the center of the image is curved to the right in the first image and curved to the left in the second image. The unpredictable deformations make the images deviate from the linear camera model assumed by optic flow algorithms. As a consequence of the aforementioned effects, the optic flow estimates are extremely noisy in comparison with typical computer vision cameras as can be used on larger robots.

The limitations may be addressed by neuromorphic visual sensors with a higher time resolution and sensitivity that are now becoming a reality [26, 24]. Instead, we have explored an alternative solution that is inspired by

¹Global shutter cameras with a reasonable energy consumption, size and weight are not yet available.



Figure 12: Two subsequent images made onboard the DelFly. The interaction between the line-by-line recording of the camera and the flapping motion leads to image deformations.

results from biology that fruit flies do not only use optic flow, but also visual appearance features for navigation [38]. Recently, we have introduced a novel appearance cue for obstacle avoidance [20], based on the following principle. When approaching an object, its colors and detailed texture become more and more visible, while other objects move out of sight. Since the color and detailed texture of one object typically vary less than the colors and textures of many different objects, the variation in colors and / or textures decreases towards impact. In [20] it has been shown that in indoor environments this *appearance variation cue* is a useful complement for detecting obstacles with the DelFly. However, the focus of that study was mainly on detection, and not on subsequent control.

In this paper we extend the autonomous flight capabilities of the DelFly II, making **two contributions**. First, an algorithm is presented for deciding to turn left or right in case of an obstacle detection. In particular, the algorithm uses the magnitude of the flow in the left and right halves of the view to determine in which way the MAV turns, as inspired by insect flight [55, 22]. The contribution is the adaptation of such an algorithm to cases in which texture is scarce. Second, the research presents the first work on the use of a pressure sensor onboard a flapping wing MAV for autonomous height control. In addition, an onboard gyro is used to better stabilize the yaw-direction of the DelFly. Since this reduces the rotational component of the optic flow, it should lead to more reliable time-to-impact estimates [55]. Onboard control of the yaw-rate should also allow for sharper turns. Below, we first explain the obstacle avoidance algorithm including the turning

logic (Subsection 4.1). Subsequently, we present experimental results on the DelFly II (Subsection 4.2).

4.1 Obstacle avoidance algorithm

In this subsection, the obstacle avoidance algorithm is explained. Obstacles are detected by a combination of the appearance variation cue (Subsection 4.1.1) and the time-to-impact as determined with optic flow (Subsection 4.1.2). In case of an obstacle detection, the turning direction is determined (Subsection 4.1.3). Finally, an overview of the control used for the avoidance experiments is given (Subsection 4.1.4).

4.1.1 Appearance variation cue

For measuring the appearance variation, the term ‘appearance’ is interpreted as texture. The approach to estimating the variation of these properties is to first estimate a probability distribution p of different textures in the image. Subsequently, the Shannon entropy $\mathcal{H}(p)$ [45] of the estimated probability distribution is calculated. For automatically determining the texture distribution, the *texton* method [51] is used. This method evaluates texture on the basis of small local image samples and was shown to outperform computationally more intensive filtering methods (e.g., Gabor filters) on a texture classification task. Below, we describe the implementation of the texton method in our experiments.

Our implementation of the texton method starts with the formation of a *dictionary* of $n = 30$ textons. To this end, small image samples of size $w \times h = 5 \times 5$ pixels are extracted from a set of images of size $W \times H = 160 \times 120$ pixels. The samples are clustered by means of a Kohonen network [32]. Please note that the dictionary in general does not need to be adapted to different environments.

After learning the dictionary, the texton method evaluates texture by estimating the probability distribution of textons in the image. s image samples are extracted from the image to build a histogram g with n bins. For each sample, the closest texton i in the dictionary is determined (Euclidian distance), and the corresponding bin in the histogram g_i is incremented. Normalizing g results in a maximum likelihood estimate \hat{p} of the texton distribution in the image, with $\hat{p}_i = g_i/s$. Please note that in computer vision applications typically all possible local samples are extracted from the

image, making \hat{p} equal to p . In the discussed experiments, a number of samples $s \ll WH$ is extracted for computational efficiency ($s = 50$). The higher computational efficiency comes at the cost of only a slightly lower accuracy (cf. [19]). The estimate \hat{p} is inserted into the formula of entropy to determine the texture variation: $\mathcal{H}(\hat{p}) = -\sum_{i=1}^n \hat{p}_i \log_2(\hat{p}_i)$. A high entropy corresponds to a high variation in appearance (assumed to be far from an obstacle), while a low entropy corresponds to the contrary (assumed to be close to an obstacle). The procedure of determining the entropy is illustrated in Figure 13.

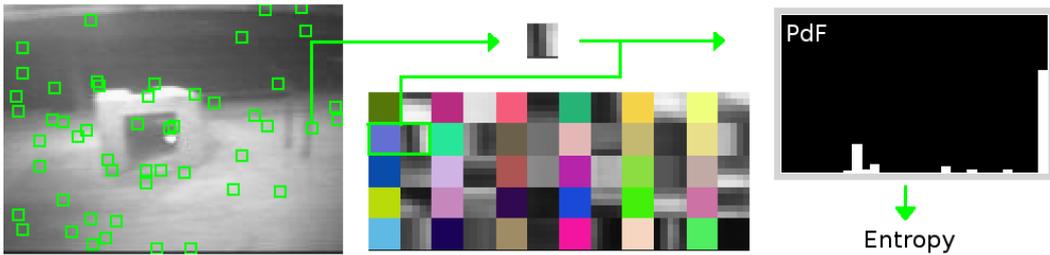


Figure 13: Procedure for determining the appearance variation: local samples are extracted from the image and matched to the textons in the dictionary. The resulting histogram is used for a maximal likelihood estimate of the texton distribution, of which the Shannon entropy is calculated.

4.1.2 Optic flow

The setting of obstacle avoidance with light-weight MAVs makes computational efficiency and robustness to noisy images the main priorities for the optic flow algorithm. For this reason, simplicity of the optic flow algorithm is preferred over detailed information on the environment.

The optic flow algorithm's implementation consists of two parts: (1) finding and tracking feature points to determine several optic flow vectors between two images, and (2) determining τ on the basis of the vectors.

For the first part, the method of Lucas-Kanade [37, 11] from the openCV library is used (<http://www.opencv.org/>). The second part is performed as follows. As in [14, 49, 39], it is assumed that the camera is moving straight towards a flat surface. The Focus of Expansion (FoE) is estimated with the least-squares method described in [49]. Subsequently, the optic flow vectors

are used to determine the distance from the old location (x_t, y_t) and the new location (x_{t+1}, y_{t+1}) to the FoE (x_e, y_e) ; $d_{e,t}$ and $d_{e,t+1}$ respectively. The difference in distance to the FoE is $\Delta d_{e,t}$. Each optic flow vector leads to one estimate $\hat{\tau}$:

$$\hat{\tau} = d_{e,t} / \Delta d_{e,t} \quad (1)$$

Since it is assumed that there is one flat surface and the optic flow vectors are noisy, the final estimate τ_{OF} is taken to be the median of the resulting $\hat{\tau}$ -distribution. The uncertainty of τ_{OF} can be captured with the standard deviation of the $\hat{\tau}$ -distribution (σ_{OF}). Despite the strong assumptions, this straightforward method works reasonably well in practice.

4.1.3 Turning logic

If the MAV is assumed to fly at a fixed height, the magnitude of the flow left and right from the FoE can be used for determining the direction in which the MAV should turn [55]. The left part of Figure 14 illustrates an MAV approaching a textured wall (stars represent textural features on the wall). If there is no rotational optic flow (or if inertial measurements can be used to derotate the optic flow), the flow in the right part of the image is stronger than that in the left part. The stronger flow indicates a shorter distance to the wall on the right, implying that the MAV should turn to the left.

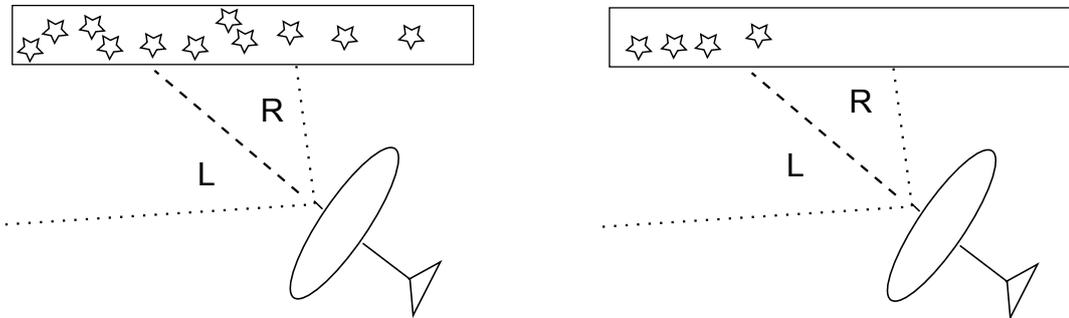


Figure 14: Left: an MAV approaches a wall with texture on both sides of the FoE. The flow can be used to determine the turning direction. Right: an MAV approaches a wall that has some texture left of the FoE and little texture on the right. Straightforward comparison of the flow magnitude results in steering towards the wall.

The above-explained turning logic has been used in previous experiments

with the DelFly II, but gave unsatisfying results. In cases where the obstacle contained little texture, as in the right part of Figure 14, the MAV would take the wrong decision and steer towards the obstacle. Such cases happen often in indoor environments, even more so when the camera field-of-view is limited (say $\approx 60^\circ$). Obviously, the matter of texture-poor obstacles has to be taken into account in the turning logic.

The main challenge for taking turning decisions is to deal with the uncertainty in the optic flow. This uncertainty derives from the sparsity of features and the noise in the optic flow measurements. The turning logic introduced here explicitly takes into account the following factors:

1. Instead of comparing flow magnitude, the local time-to-impact estimates $\hat{\tau}$ are averaged on each side of the FoE. Consequently, $\tau_{RL} = \tau_R - \tau_L$ is the measure used for determining which way to turn: if the time-to-impact is larger on the right (τ_{RL} positive), the DelFly should turn to the right.
2. The number of features on the right and the left sides of the FoE (f_R , f_L) is an indication of the uncertainty of the τ_{RL} estimate. Fewer features on one of the sides more probably lead to erroneous estimates of τ_{RL} .
3. The suitability of the texture for determining optic flow and consequently time-to-impact can be automatically evaluated. The optic flow algorithm selects features at locations with high ‘quality’, i.e., where the minimal eigenvalue of the derivative covariation matrix is high with respect to neighboring locations. Averaging the minimal eigenvalue over the image is a good indication of the uncertainty in the optic flow estimates. This average is henceforth referred to as q .
4. Under the assumptions of a static environment and forward flight, τ_L and τ_R should always be positive. Negative values are considered erroneous since they imply either other moving objects or backward motion.
5. A single estimate of τ_{RL} is likely not sufficient to determine a turning direction. Insufficient texture or all kinds of noise can render the estimate unreliable. Therefore, information should be integrated over time.

The factors above are incorporated in an algorithm that integrates information on τ_{RL} and its uncertainty over time. Throughout a flight, the DelFly Turning Algorithm (DTA) maintains a turning decision value D_{RL} that is used upon an obstacle detection to determine the turning direction. Positive D_{RL} values represent turns to the right, while negative values represent turns to the left. Since it may happen that no evidence is gathered on the turning direction before the first obstacle detection, D_{RL} is given an initial value of 1, biasing the first turn to the right. At the start, this decision is without evidence ($e = 0$). Subsequently, per image pair DTA first verifies that (i) the DelFly is not currently performing a turn (as measured by the median horizontal optic flow per second, $u_x < T_{ux}$), (ii) the image pair contains features of sufficient quality ($q \geq T_q$), (iii) the number of features on each side is not too low ($\min(f_R, f_L) \geq T_f$), and that (iv) the time-to-contact estimates τ_L and τ_R are not negative. If on the basis of this verification, τ_{RL} is considered sufficiently reliable, the decision value D_{RL} is updated as follows:

$$D_{RL} \leftarrow \frac{(eD_{RL} + e'\tau_{RL})}{e + e'}, \quad (2)$$

where

$$e' = w_q q + w_f \min(f_L, f_R), \quad (3)$$

implying that the decision is a weighted average of τ_{RL} estimates.

An initial guess of the parameter settings (thresholds and weights) has been made on the basis of the typical magnitudes of the involved variables. Subsequently, an iterative process was followed of testing, analysis, and (manual) refinement of the parameter settings in simulation. For brevity we do not report in detail on the simulation results. Still, it is noteworthy that in simulation, the average time before a crash when deciding the turn on the basis of the instantaneous τ_{RL} is 62 s. The average time before a crash with DTA is 162 s, an improvement of 100 s. The parameters used in simulation were then further tuned for the real system. The resulting parameter settings are: $T_{ux} = 55$ pixels / sec (in 160×120 pixel images with a horizontal FOV of 61.3°), $T_q = 10^{-4}$, $T_f = 10$, $w_q = 5000$, and $w_f = 0.1$. A pseudo-code representation of DTA is shown as Algorithm 1.

4.1.4 Overview of the control algorithm

There are two control loops that are separate from each other: the height control is performed by the microcontroller onboard the DelFly on the basis

Algorithm 1 DelFly Turning Algorithm (DTA)

```
1:  $D_{RL} \leftarrow 1, e \leftarrow 0$ ;  
2: while flying do  
3:   Determine optic flow and  $\tau_{RL}$   
4:   if Commanding turn  $\vee |u_x| > T_{ux}$  then  
5:      $e \leftarrow 0$   
6:   else if  $q \geq T_q \wedge \min(f_R, f_L) \geq T_f \wedge \tau_L, \tau_R \geq 0$  then  
7:      $e' \leftarrow w_q q + w_f \min(f_L, f_R)$   
8:      $D_{RL} \leftarrow \frac{D_{RL}e + \tau_{RL}e'}{e + e'}$   
9:      $e \leftarrow e + e'$   
10:  end if  
11: end while
```

of the pressure sensor, while the flying direction is determined by a laptop that serves as ‘ground station’.

For the obstacle avoidance experiments the height is controlled by the microcontroller onboard of the DelFly (an Atmega88PA). The SCP1000 barometer measures the pressure in the flight room at a frequency of 5 Hz to a resolution of 5 Pa which corresponds to about 40 cm. Using significant low-pass filtering (which is acceptable thanks to the slow and well damped vertical dynamics of DelFly) the resolution of this measurement with approximately Gaussian noise can be highly increased. The pressure as measured at the start of the experiment is used as a reference value. PI-control with saturated integrator input and low pass filtered proportional term is employed for regulating the height, which can be done by increasing or decreasing thrust. As is well-known, the pressure can vary significantly over longer time spans, but currently such time spans are longer than the flight duration of the DelFly.

The ground station determines the flight direction by giving rudder commands to the microcontroller onboard the DelFly. These commands are: fly straight, turn to the left, and turn to the right. When a turn is commanded, the microcontroller sets a yaw reference value at $\pm 90^\circ$. It then integrates the yaw rate measurements over time, actuating the rudder to perform the turn. The algorithm for deciding upon the rudder command to give is illustrated with a flow-chart in Figure 15. In case of a collision detection, the value of D_{RL} determines the turning direction. Then

the turn is commanded, while suppressing new turns for a predetermined ‘turn time’². The turning sequence is always executed fully before new commands are allowed. When not executing a turn, a collision is detected if: $(\tau_{OF} \geq 0 \wedge \tau_{OF} < 2 \wedge \sigma_{OF} < 2 \wedge |\dot{\psi}| < 12^\circ/\text{sec}) \vee \mathcal{H}(\hat{p}) < 2.2$. The values for the thresholds have been determined in simulation and subsequently tuned on the real platform [20]. The condition on $|\dot{\psi}|$ is used for preventing new detections during a turn. A cascaded PID-controller is used to control the lateral dynamics of DelFly. The innerloop consists of a P-controller on the turn-rate to damp the yawing motions and prevent spiral motions. The outerloop consists of a PID-controller on the integrated gyro signal. While this signal slightly drifts over time, the controller results in rather straight flight trajectories, even in the presence of disturbances.

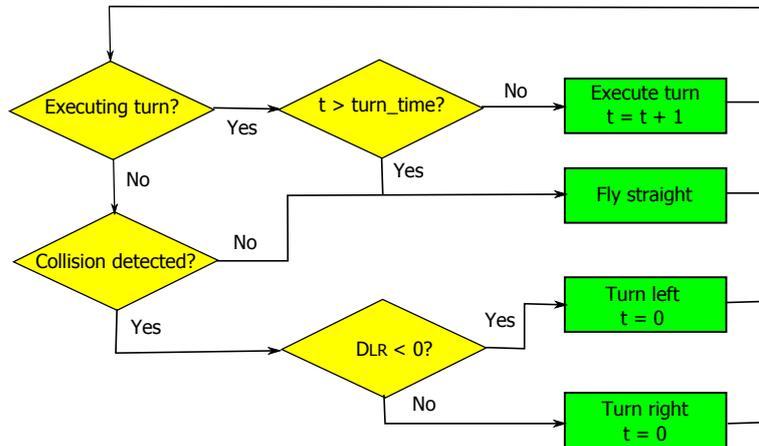


Figure 15: Flowchart for the control of the rudder. See the text for details.

While the vision algorithms are currently running offboard, their efficiency remains of uttermost importance in order to control small MAVs with fast dynamics and to have a chance of being implemented onboard in the near future. The video frame rate is 30Hz. The frames are down-sized to 160×120 images and processed by both the optic flow algorithm running

²Since there is no communication yet from the microcontroller to the ground station, the ground station uses a fixed time instead of a signal that indicates turn completion. This fixed time has been tuned manually for an average turning time. Slight underestimates of the turn are not so problematic, since the collision detection takes the median horizontal flow u_x into account. Small overestimates typically give no problem, unless the turn brings the DelFly very close to another obstacle.

at 30Hz and the texton method running at 13Hz. The execution frequencies have been measured on a dual-core 2.26GHz laptop with all ground station software running at the same time.

4.2 Results obstacle avoidance experiments

The obstacle avoidance experiments performed for this study have as goal to test the turning logic in the presence of texture-poor obstacles. The test room is an office room with a flyable area of $\sim 3.20 \times 2.70 \times 2.80$ m. Autonomous flight in such a relatively small room is challenging since there are many obstacle encounters. In particular, given the typical flight speed there is a short time window for obstacle detections. On the one hand, the flapping wing MAV has to detect the obstacles on time for turning before a collision (> 1.5 s). On the other hand, too long a detection time (> 3.0 s) would already result in continuous turning. Below, we analyze the data from an experiment that illustrates both the strengths and limits of the DTA discussed in Subsection 4.1.3 - selecting a test that ends in a collision. Videos of this and other experiments can be found online ³.

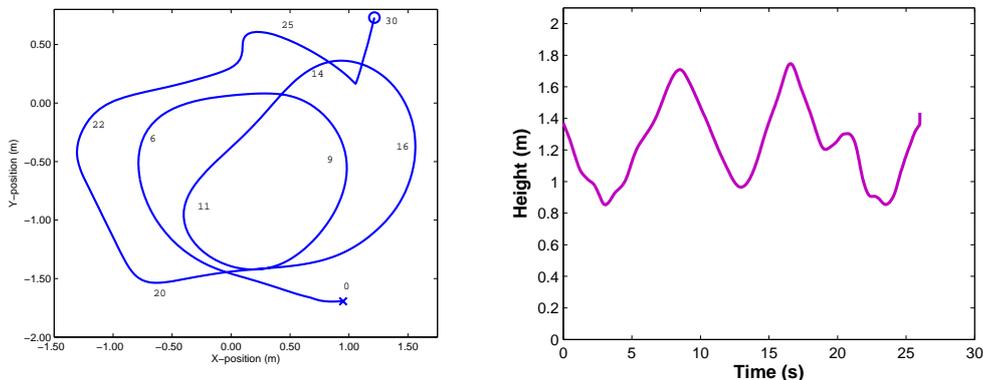


Figure 16: Left: (x, y) -position of the DelFly II over time during one of the experiments. The top, left, and right lines of the box illustrate the walls in the room, the bottom line the table with the ground station. Right: height of the DelFly during the test.

The (x, y) -positions of the DelFly during the flight are shown in the left

³http://www.youtube.com/watch?v=_p6a8ei4PZc

part of Figure 16, where the cross marks the start of autonomous control and the circle the end of the trial. The left, top, and right lines of the box illustrate a wall with a couch, a wall with a whiteboard, and windows with closed blinds, respectively. The bottom line represents the table with the ground station. Closely behind the table are the experimenters and a wall. As a consequence of this setup, the DelFly also avoids the bottom limit of the flight area. The right part of Figure 16 shows the height of the DelFly over time.

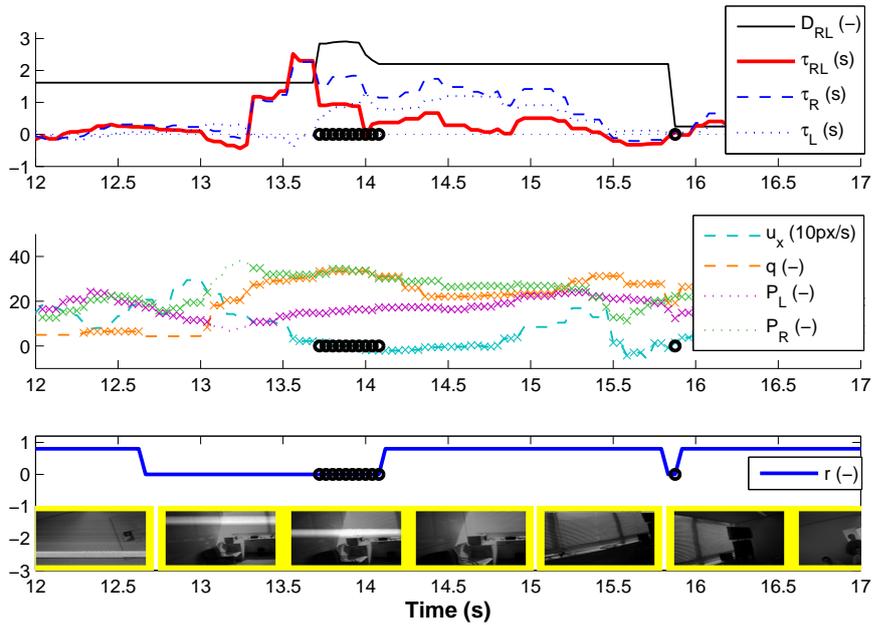


Figure 17: All signals relevant to the turning logic (best viewed in color, see text for details). Units are indicated between brackets, with (-) if the signal does not have a unit. The turning logic ensures that the decision variable D_{RL} is not updated if the estimate of τ_{RL} is unreliable. For example, around 12 s the images have little texture, while around 13 s the DelFly is turning. This leads to bad estimates that are successfully filtered out. Time steps at which the decision value is updated are indicated with black circles in all three plots.

From the experiments, we make four main observations. The first observation concerns the height control with the pressure sensor: in spite of the

low-frequency and low-accuracy measurements, the PI-controller successfully controls the DelFly’s height within safe margins. The variation in the height is ± 40 cm around the set point. Much of this variation seems to be caused by the rather abrupt turns, since the variation is much smaller when following a smoother trajectory (approximately ± 20 cm). The average height over the experiment does not vary over the time span of this experiment. In addition, if there were any sudden changes in pressure they have been filtered out successfully⁴. Importantly, the local pressure variations due to the flapping of DelFly are not significant anymore after application of the PI controller’s low-pass filter. The height controller can be switched on before flight and still functions correctly when in hover flight.

The second observation is a confirmation of the findings from the literature (e.g., [55]) that the turning direction tends to remain the same over time in a rectangular room. Often the DelFly turns right as this is the initial value given to D_{RL} . Indeed, if we take a look at the trajectory in Figure 16, all turns until the last one are to the right. This is a consequence of the followed turning strategy and the rectangular layout of the rather small room.

The third observation is that DTA’s integration of information over time and use of the reliability of time-to-impact estimates is absolutely necessary to take correct turning decisions. The three plots of Figure 17 show all the signals related to the turning logic over time. The top plot contains D_{RL} (black line), τ_{RL} (red), τ_R (dark blue, dashed), τ_L (dark blue, dotted). The middle plot shows the following variables, some scaled with constant factors for visualization purposes: the number of points left and right, p_L (purple dotted), p_R (green dotted), the quality q of the features (orange dashed, scaled by 10^4), the heading flow u_x (light blue, scaled by 0.1). Time steps at which these signals allow for an update of D_{RL} are indicated with a cross. The bottom plot shows the rudder command r (blue) and the onboard images centered on their corresponding time step. In all plots, black circles are plotted when the conditions allow for an update of the decision value D_{RL} .

The most important signal is τ_{RL} , the difference in time-to-impact between the right and left side of the FoE. Its value is often wrong. Around

⁴In preliminary tests with the height regulation, sudden pressure changes were induced by opening and closing a door to the flying room. Thanks to the slow vertical dynamics of DelFly, this did not have a strong impact on the flying height.

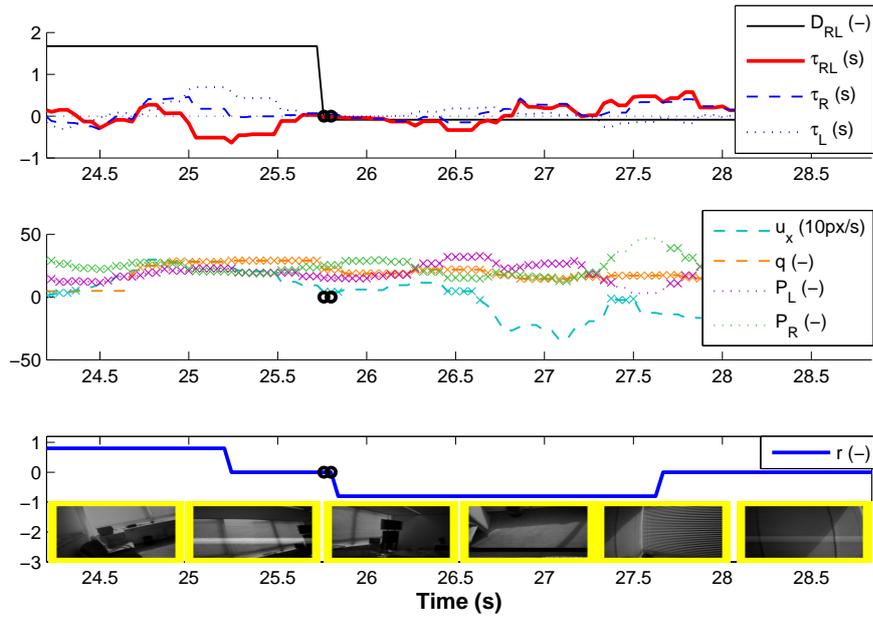


Figure 18: All signals relevant to the turning logic (best viewed in color, see text for details). Units are indicated between brackets, with (-) if the signal does not have a unit. Noisy images can lead to bad estimates. In combination with the resetting of the turning evidence e after a turn, this may lead to wrong turning decisions. This occurs around 25.8 s, resulting in a turn to the left that brings the DelFly into a corner from which it cannot exit in time.

12 s it is slightly negative as a consequence of little texture and around 15.7 s it is negative due to the execution of a turn. These wrong values are respectively detected by means of $q < T_q$ (little texture), and a large heading flow (plus active rudder command). As a consequence of ignoring the negative values, the DelFly does not turn left when detecting an obstacle at 15.9 s.

The fourth observation is that with the current speed and the confined space in which the experiment is performed, the DelFly spends a large time executing turns. While executing turns, the evidence e is reset and no new evidence is gathered. As a consequence, it often happens that the DelFly bases its turning decision on a few measurements. If these measurements are mistakenly considered ‘reliable’, they can have a detrimental effect on the turning decision D_{RL} . Figure 18 shows how noise in the image leads to a wrong estimate of τ_{RL} just after a turn (~ 25.8 s). Because there is an obstacle detection, the DelFly turns to the left immediately, bringing it into a corner from which it cannot turn away in a timely fashion.

5 Conclusion

The DelFly is still a long way from a fly-sized autonomously flying robot. Nonetheless, the presented findings from combined aerodynamic and autonomy experiments illustrate some of the key properties necessary for such a flapping wing robot.

With respect to the DelFly’s design, the influence of the different design parameters on the thrust-to-power ratio of the DelFly II are likely to still play a role on a smaller scale. This was already experienced with the DelFly Micro, that features a double clap-and-peel on the basis of findings that this significantly contributes to lift generation. The presented experiments resulted in an improvement in the generated lift, allowing the inclusion of a yaw rate gyro, pressure sensor, and microcontroller onboard the DelFly. An experimental aerodynamic investigation provided evidence of a conical leading edge vortex development during the flap cycle, where inside the vortex tube a spanwise velocity component outward was found. Future research on the design could focus on a resonant flapping-wing mechanism (possibly as in [6]), since it would further improve the energy efficiency. Future work could also include actively controlling multiple degrees of freedom of the wings, allowing flight stabilization without a tail (as on the Aerovironment Hummighbird [1]).

With regards to the DelFly’s autonomous flight capabilities, a step forward was made both in height control and obstacle avoidance. The height control was successfully performed by the onboard microcontroller using a pressure sensor. The pressure differences due to the flapping flight did not have a detrimental effect on height control. Still, the height variations were in the range of $\pm 40\text{cm}$, which means that there is room for improvement. Sensors with better resolution or update rate, or even additional information from accelerometers or vision may help to reduce the height variations and may provide a means to keep a safe height also for longer time spans. Concerning the obstacle avoidance, a step forward was made by introducing an improved turning logic. Together with onboard control of the yaw rate, this led to improved obstacle avoidance capabilities with respect to instantaneous use of the flow magnitude in a confined office space. Still, faulty decisions occur, resulting in collisions with indoor obstacles. One small improvement of the DTA may be to not reset the evidence e to 0, reducing the influence of only a few τ_{RL} measurements. As main directions for larger improvements of the obstacle avoidance capabilities, we identify the incorporation of onboard vision processing for reducing noise and allowing the direct feedback of gyro measurements (cf. [55]) or the feedback current from the motors (as in [7]) for derotating optic flow. In addition, work on shorter turn radiuses of DelFly would also be of great help in confined spaces such as small office rooms. Finally, neuromorphic vision sensors with a higher update frequency, sensitivity, and a larger field-of-view may lead to further improvements of obstacle avoidance capabilities.

References

- [1] Aerovironment inc.: Nano hummingbird - flapping wing mav. <http://www.avinc.com/nano>, 2011.
- [2] R.J. Adrian, K.T. Christensen, and Z.C. Liu. Analysis and interpretation of instantaneous turbulent velocity fields. *Experiments in Fluids*, 29(3):275–290, 2000.
- [3] S. Ahrens. Vision-based guidance and control of a hovering vehicle in unknown environments. Master’s thesis, MIT, 2008.

- [4] H. Aono, S.K. Chimakurthi, C.E.S. Cesnik, H. Liu, and W. Shyy. Computational modeling of spanwise flexibility effects on flapping wing. In *47th AIAA Aerospace Sciences Meeting*, 2009.
- [5] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *EMAV, the Netherlands*, 2009.
- [6] S.S. Baek, K.Y. Ma, and R.S. Fearing. Efficient resonant drive of flapping-wing robots. In *IROS 2009*, pages 2854–2860, 2009.
- [7] F. Garcia Bermudez and R. Fearing. Optical flow on a flapping wing robot. In *IROS 2009*, pages 5027–5032, 2009.
- [8] A. Beyeler, J-C. Zufferey, and D. Floreano. 3d vision-based navigation for indoor microflyers. In *ICRA 2007, Italy*, pages 1336–1341, 2007.
- [9] J.M. Birch, W.B. Dickson, and M.H. Dickinson. Force production and flow structure of the leading edge vortex on flapping wings at high and low reynolds numbers. *The journal of Experimental Biology*, 207(7):1063–1072, 2004.
- [10] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *IEEE International Conference on Robotics and Automation*, 2010.
- [11] Jean-Yves Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. description of the algorithm, 2000.
- [12] N.L. Bradshaw and D. Lentink. Aerodynamic and structural dynamic identification of a flapping wing micro air vehicle. In *26th AIAA Applied Aerodynamics Conference, Honolulu, Hawaii*, 2008.
- [13] R. A. Brooks. Elephants don’t play chess. *Robotics and Autonomous Systems*, 6(1-2):3–15, 1990.
- [14] T. Camus. Calculating time-to-contact using real-time quantized optical flow. Technical report, Max Planck institute for Biological Cybernetics, TR-14, 1995.
- [15] K. Celik, S.J. Chung, and A. Somani. Mono-vision corner slam for indoor navigation. In *(EIT 2008)*, pages 343–348, 2008.

- [16] T.S. Collett. Insect vision: Controlling actions through optic flow. *Current Biology*, 12:615–617, 2002.
- [17] A.J. Davison and D.W. Murray. Simultaneous localisation and map-building using active vision. *IEEE PAMI*, 2002.
- [18] G.C.H.E. de Croon, K.M.E. de Clerq, R. Ruijsink, B. Remes, and C. de Wagter. Design, aerodynamics, and vision-based control of the delfly. *International Journal on Micro Air Vehicles*, 1(2):71 – 97, 2009.
- [19] G.C.H.E. de Croon, C. de Wagter, B.D.W. Remes, and R. Ruijsink. Random sampling for indoor flight. In *International Micro Air Vehicle conference, Braunschweig, Germany (2010)*, 2010.
- [20] G.C.H.E. de Croon, E. de Weerdt, C. de Wagter, B.D.W. Remes, and R. Ruijsink. The appearance variation cue for obstacle avoidance. *IEEE Transactions on Robotics*, in press.
- [21] M.H. Dickinson, F.-O. Lehmann, and S.P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284(5422):1954–1960, 1999.
- [22] M. Egelhaaf and R. Kern. Vision in flying insects. *Current Opinion Neurobiology*, 12(6):699–706, 2002.
- [23] C.P. Ellington, C. van den Berg, A.P. Willmott, and A.L.R. Thomas. Leading-edge vortices in insect flight. *Nature*, 384(19/26):626–630, 1996.
- [24] F. Expert, S. Viollet, and F. Ruffier. Outdoor field performances of insect-based visual motion sensors. *Journal of Field Robotics*, 28(4):529–541, 2011.
- [25] N. Franceschini, J.M. Pichon, C. Blanes, and J.M.Brady. From insect vision to robot vision. *Philosophical Transactions: Biological Sciences*, 337(1281):283–294, 1992.
- [26] N. Franceschini, S. Viollet, F. Ruffier, and J. Serres. Neuromimetic robots inspired by insect vision. *Advances in Science and Technology*, 58:127–136, 2008.
- [27] L.J. Goodman. The landing responses of insects I. the landing response of the fly, *Lucilia sericata*, and other calliphoridae. *Journal of Experimental Biology*, 37:845–878, 1960.

- [28] M.A. Groen, B. Bruggeman, B.D.W. Remes, R. Ruijsink, B.W. van Oudheusden, and H. Bijl. Improving flight performance of the flapping wing mav delfly ii. In *International Micro Air Vehicle conference, Braunschweig, Germany (2010)*, 2010.
- [29] S. Grzonka, G. Grisetti, and W. Burgard. Towards a navigation system for autonomous indoor flying. In *(ICRA 2009), Kobe, Japan*, 2009.
- [30] A.M. Hyslop and J.S. Humbert. Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow. *AIAA Guidance, Control, and Dynamics*, pages 147–159, 2010.
- [31] F. Iida. Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision. In *(ISR 2001)*, 2001.
- [32] T. Kohonen. *Self-Organizing Maps, third edition, Springer Series in Information Sciences, Vol. 30*. Springer, 2001.
- [33] F.-O. Lehmann. The mechanisms of lift enhancement in insect flight. *Naturwissenschaften*, 91(3):101–122, 2004.
- [34] F.-O. Lehmann. Review: When wings touch wakes: understanding locomotor force control by wake-wing interference in insect wings. *The journal of Experimental Biology*, 211(2):224–233, 2008.
- [35] S. Leven, J.-C. Zufferey, and D. Floreano. A minimalist control strategy for small UAVs. In *(IROS 2009)*, pages 2873–2878, 2009.
- [36] M.J. Lighthill. On the Weis-Fogh mechanism of lift generation. *J. Fluid Mech.*, 60(1):1–17, 1973.
- [37] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.
- [38] G. Maimon, A.D. Straw, and M.H. Dickinson. A simple vision-based algorithm for decision making in flying drosophila. *Current Biology*, 18(6):464–470, 2008.
- [39] M. Makhoulta, Y. Gao, and K. Shala. A vision and behaviour based approach for short-range autonomous navigation of planetary rovers.

In *ESA Workshop on Advanced Space Technologies for Robotics and Automation - ASTRA*, Noordwijk, Netherlands, 2008.

- [40] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press / Bradford Books, Cambridge, MA, 2000.
- [41] R. Pfeifer and C. Scheier. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.
- [42] F.C. Rind. Motion detectors in the locust visual system: from biology to robot sensors. *Microscopy research and technique*, 56:256–269, 2002.
- [43] F. Ruffier and N.H. Franceschini. Aerial robot piloted in steep relief by optic flow sensors. In *(IROS 2008)*, pages 1266–1273, 2008.
- [44] S.P. Sane. Review: The aerodynamics of insect flight. *The journal of Experimental Biology*, 206(23):4191–4208, 2003.
- [45] C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.
- [46] M.V. Srinivasan, M. Poteser, and K. Kral. Minireview: Motion detection in insect orientation and navigation. *Vision Research*, 39:2749–2766, 1999.
- [47] E. Steltz, S. Avadhanula, and R.S. Fearing. High lift force with 275 hz wing beat in mfi. In *IEEE International Conference on Intelligent Robots and Systems*, 2007.
- [48] E. Steltz and R.S. Fearing. Dynamometer power output measurements of piezoelectric actuators. In *IEEE International Conference on Intelligent Robots and Systems*, 2007.
- [49] N. Takeda, M. Watanabe, and K. Onoguchi. Moving obstacle detection using residual error of FOE estimation. In *IROS*, pages 1642–1647, 1996.
- [50] C. van den Berg and C.P. Ellington. The three-dimensional leading edge vortex of a 'hovering' model hawkmoth. *Phil. Trans. R. Soc. Lond. B*, 352(1351):329–340, 1997.

- [51] M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *(CVPR 2003)*, volume 2, pages 691–698, 2003.
- [52] H. Wagner. Flow-field variables trigger landing in flies. *Nature*, 297:147–148, 1982.
- [53] P.W. Willmott, C.P. Ellington, and A.L.R. Tomas. Flow visualization and unsteady aerodynamics in the flight of the hawkmoth, *manduca sexta*. *Phil. Trans. R. Soc. Lond. B*, 352(1351):303–316, 1997.
- [54] R.J. Wood. The first takeoff of a biologically-inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2):341–347, 2008.
- [55] J.-C. Zufferey. *Bio-inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers*. Lausanne : EPFL/CRC Press, 2008.