Autonomous Flight of Flapping Wing Micro Air Vehicles

Tijmons, Sjoerd

DOI
10.4233/uuid:74fee365-ba6d-456a-8ec0-358dc708eef4

Publication date
2017

Document Version
Final published version

Citation (APA)
Tijmons, S. (2017). *Autonomous Flight of Flapping Wing Micro Air Vehicles*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:74fee365-ba6d-456a-8ec0-358dc708eef4

Autonomous Flight of Flapping Wing Micro Air Vehicles

Autonomous Flight of Flapping Wing Micro Air Vehicles

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 12 december 2017 om 15:00 uur

door

Sjoerd Tijmons

Master of Science in Aerospace Engineering,
Technische Universiteit Delft, Delft, Nederland,
geboren te Rotterdam, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. M. Mulder
copromotor: dr. G.C.H.E. de Croon

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Mulder ,	Technische Universiteit Delft, promotor
Dr. G.C.H.E. de Croon ,	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Dr. G.W. Kootstra,	Wageningen Universiteit
Prof. dr. M. Kovac,	Imperial College London
Prof. dr. D.M. Gavrila,	Technische Universiteit Delft
Prof. dr. ir. M. Wisse,	Technische Universiteit Delft
Prof. dr. H.C. Park,	Konkuk University, Seoul

TU Delft Delft University of Technology

Keywords: Stereo Vision, Obstacle Avoidance, Micro Air Vehicles, Flapping Wings, Robotics, Autonomous Flight, Indoor Flight, Self-Supervised Learning, Attitude Estimation And Control, Wing Actuation Mechanism

Printed by: Gildeprint, Enschede

Front & Back: Image: DelFly Explorer, Design: Sjoerd Tijmons

ISBN 978-94-6233-834-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

To Jessica

for your love, your support and your patience

Summary

Many types of drones have emerged over the last decade and new applications in various sectors are announced almost on a daily basis. In scientific literature, small drones are called Micro Air Vehicles (MAVs). Especially very small MAVs will play a significant role in indoor applications, since their small size allows them to navigate in narrow, cluttered environments. At the same time, many indoor applications will benefit from MAVs becoming fully autonomous. That will allow these vehicles to operate in areas that cannot be accessed by humans for various reasons.

However, these promising small and lightweight MAVs still have very limited autonomous flight capabilities, mainly due to weight restrictions. Since MAVs need to lift their own weight, the sensing and processing devices that can be taken on board is limited. The size, weight and power (SWaP) characteristics of such components all influence the total required and total available power for flight. To enable autonomous flight of very small MAVs, it is therefore essential to select a combination of sensors and robust algorithms that form an effective trade-off between accuracy and SWaP characteristics.

For an MAV to fly fully autonomously, it needs a combination of various capabilities. For characterizing the autonomy of an MAV, such capabilities are subdivided into different levels of autonomy. These levels range from low-level capabilities to high-level capabilities: (1) attitude control, which is needed for performing stable flight, rejecting disturbances and performing agile maneuvers, (2) height control, the ability to control the vertical speed and the altitude, (3) collision avoidance, which in essence is about maintaining a safe distance to detected obstacles, and (4) navigation, which involves many capabilities that relate to deciding where the vehicle can go or should go, and also where the vehicle has been.

Considerable progress has been made in achieving autonomous tasks with large MAVs for all these levels. However, for small and lightweight MAVs only very limited capabilities were demonstrated so far. At the onset of this work, no solution had been demonstrated that combines even the first three levels levels of autonomous flight on a small and lightweight MAV.

This study specifically focuses on the very lightweight class of flapping wing MAVs. Nature has many examples showing that the flapping principle can be used at very small scales. Studies have revealed that some insect wings can produce three times more lift than would be expected based on conventional aerodynamic effects. At insect-scales, aerodynamic forces are likely to be produced more efficiently by flapping motions than by rotational motions. A unique feature of flapping wings is also that they often combine the generation of lift forces, thrust forces and control moments. For these reasons it is an interesting concept to explore in the process of miniaturizing MAVs. Even though the research interest in flapping wing MAVs is growing, the number of studies that focus on autonomous flight of such vehicles

is rather limited. The majority of these studies that have been performed focus on the first two flight levels only. No work has been published so far where the first three or four levels of autonomous flight are combined on a flapping wing MAV. Therefore the following research goal is formulated:

> **Research Goal**: Develop autonomous flight capabilities for lightweight flapping wing Micro Air Vehicles.

To reach this goal, three research questions were posed. As a starting point of this research, stereo vision is selected as the primary sensor to be carried on board the vehicle. In this study, the DelFly is selected as the flapping wing platform. The first research question focuses on this combination of sensor and platform:

> **Research Question 1**: How can stereo vision be used on flapping wing Micro Air Vehicles for autonomous flight tasks?

The first step in answering this question is testing whether stereo vision works well enough on a flapping wing MAV for obstacle avoidance, and whether it outperforms the more common method of using optical flow. A 5.2 gram stereo vision system is added to the DelFly flapping wing platform. It sends analogue video to a ground station that runs a stereo vision algorithm and produces heading control commands which are sent back to the vehicle. A new obstacle avoidance method is proposed that is expected to provide collision-free avoidance maneuvers at any time, in any situation. Collision-free flights were recorded with this system of 21 gram with durations up to 72 seconds. These tests show the great potential of using stereo vision for obstacle detection and avoidance, even though this system relies on off-board processing.

Fully onboard integration of stereo vision on the DelFly flapping wing MAV is therefore realized and tested as a second step. A custom-made stereo vision system of 4 grams is introduced which includes all functionality to allow onboard obstacle detection and avoidance. A new vehicle design is introduced, called "DelFly Explorer". It comprises a new tail design and additionally active aileron surfaces, which were found to be essential based on the earlier tests that included off-board processing. Flight times up to 9 minutes have been recorded with this 20 gram system, limited by the battery capacity. A stereo vision algorithm is introduced called "LongSeq" which is suitable for running on this system in realtime. Decision making for obstacle avoidance is based on a reactive control algorithm. This approach has been tested in flight tests with sparse obstacle fields, where the obstacles are separated sufficiently to allow space for evasion.

The obstacle avoidance method developed in the first step is further extended in the third step. This avoidance strategy, called "Droplet", is applied to the DelFly Explorer. This method is better suited for flying in complex environments, such as dense obstacle fields and environments with walls. A different stereo vision algorithm is proposed that leads to more robust avoidance performance in combination with the Droplet strategy. Depending on the type of environment, flight times of 9 minutes without collisions can be achieved by this approach.

An alternative way to maximize the performance and robustness of the vision system is by learning from earlier observations. To this end a second research question is posed:

> **Research Question 2**: How can a monocular robot learn by itself to see distances to obstacles by means of appearance?

A method is developed which involves appearance-based learning of distances to objects. A monocular camera system of 2 grams is used in this setup. In combination with low weight proximity sensors for near-collision detection, this method learns the appearance extracted from the camera images in situations where the vehicles approaches nearby objects. Since the memory available on board the camera system can only store a small number of images, an efficient image description algorithm is used to compress the image data. This is essential for the learning process, which requires sufficient training data. It was found that a standard k-NN approach is effective only when a large amount of training samples is stored, which is not feasible on the camera system used in this study. A clustering step is therefore included in the training process that maximizes the amount of stored training data while stimulating that the variation in image appearance types remains high. Computer simulation results show that in confined spaces the learning performance after including the clustering step is only marginally reduced. The simulations also show that by using the learning data for estimating distances from new image frames allows the vehicle to prevent near-collisions from happening. Real flight tests with the 19 gram vehicle indicate that the learning rate is lower compared to the computer simulation results, but similar trends are still observed. By storing training data over several flights, the vehicle is able to perform flights with successful obstacle avoidance in a confined room.

The first two research questions have a main focus on using sensors and algorithms to perform autonomous flight tasks. However, another approach would be to adapt the robot's body to facilitate its autonomous flight capabilities. Therefore the third research question is formulated as follows:

> **Research Question 3**: Can the performance of the obstacle avoidance task and other navigation tasks be improved by increasing the control authority of flapping wing Micro Air Vehicles?

An innovative control mechanism is proposed which is primarily intended to improve the heading control performance of the flapping wing MAV when in hover. Especially when obstacles are detected too late to perform a standard avoidance turn, the ability to hover and turn around in approximately the same position would be beneficial as it provides an alternative and perhaps very versatile and robust escape maneuver. Flight tests show that indeed high heading rates can be achieved and that the 21 gram vehicle is able to reverse its flight direction while requiring only a small turn space. The proposed mechanism allows to control the pitch and roll angles as well. Hence, despite the cost of using an extra servo and heavier/stronger servos, the attitude of the vehicle can be controlled fully without the need of a tail, resulting in a system weight of just under 20 gram.

To conclude, fully onboard implementations of the first three autonomous flight levels have been realized on a flapping wing MAV: attitude control, height control and obstacle avoidance. For attitude control a new wing control mechanism has been realized, which extends the flight envelope to controlled hovering flight and sideways flight. This enhances the obstacle avoidance capabilities by enabling turning using a very small space. Height control by barometric sensor feedback has been realized, providing a reasonable performance. Stereo vision is shown to be a feasible and very effective solution to performing obstacle avoidance. It considerably outperforms optical flow based solutions. Obstacle detection and avoidance based only on monocular vision through learning is found to be feasible in a limited space. For the development of the autonomous capabilities in this study it was found to be very important to take into account the relationships between sensing, processing, vehicle design, vehicle behavior and the environment. An experimental approach is important for finding all these relationships.

For the fourth level of autonomous flight, navigation, valuable insights have been obtained. A potential navigation capability would be to maximize the visited area during a flight by preventing the vehicle from flying repetitive patterns. The (appearance) learning principle is regarded as an effective approach for recognizing earlier visited places. The new flight capabilities, such as hovering and flying sideways, allow to perform specific tasks in narrow spaces, such as flying through narrow corridors and opened doors and windows. This enables new maneuvers and different behaviors.

Future work may be performed in three main directions. The first part is the integration of the stereo vision system developed in the first chapters with the vehicle equipped with the control mechanism proposed in Chapter 6. The second part is the development of smaller versions of the flapping wing MAV with the same autonomous capabilities, by realizing a smaller weight of the autopilot board and stereo vision system. The third part is extending the current autonomous capabilities with more advanced navigation capabilities. First steps in this process can be defined based on insights obtained in this thesis. The principle of learning the appearance of the environment might also be used for recognizing earlier visited locations. Recently developed Time-of-Flight sensors are expected to enable various autonomous flight capabilities at the cost of adding just a small additional weight. These can be useful to perform additional tasks such as wall following, door detection, narrow corridor traversal, and distance estimation to texture-poor surfaces and transparent windows. In combination with the current stereo vision system, these sensors would enable autonomous flight on all four levels.

Samenvatting

De afgelopen tien jaar zijn er veel verschillende soorten drones ontwikkeld en nieuwe toepassingen in verscheidene sectoren worden bijna dagelijks aangekondigd. In wetenschappelijke literatuur worden kleine drones aangeduid als Micro Air Vehicles (MAV's), ofwel micro-luchtvaartuigen. Vooral heel kleine MAV's zullen een belangrijke rol spelen in binnentoepassingen, daar hun kleine afmetingen het mogelijk maken om in nauwe, onoverzichtelijke ruimtes te navigeren. Tegelijkertijd is het voor veel binnen-toepassingen voordelig als MAV's autonoom opereren. Dat maakt het mogelijk om deze toestellen te gebruiken in ruimtes die om uiteenlopende redenen niet toegankelijk zijn voor mensen.

Deze veelbelovende kleine en lichte MAV's hebben momenteel echter nog zeer beperkte autonome mogelijkheden, voornamelijk door gewichtsrestricties. Omdat MAV's hun eigen gewicht moeten tillen, zijn de sensor- en verwerkingssystemen die kunnen worden ingebouwd erg beperkt. De grootte, het gewicht en stroomverbruik van zulke componenten zijn nadelige karakteristieken die allemaal invloed hebben op het vermogen dat overblijft om te vliegen. Om autonoom vliegen mogelijk te maken voor MAV's, is het daarom essentieel om een combinatie van sensoren en robuuste algoritmes te selecteren die een effectieve afweging vormt tussen nauwkeurigheid enerzijds, en die nadelige karakteristieken anderzijds.

Om autonoom te vliegen moet een MAV verschillende taken tegelijk uitvoeren. Om de autonome mogelijkheden van een MAV te classificeren worden die taken ingedeeld in vier niveaus van autonomie. Deze niveaus lopen op van laag naar hoog: (1) standregeling, wat nodig is om stabiel te vliegen, met verstoringen om te gaan en om behendig te manoeuvreren, (2) hoogtebesturing, de mogelijkheid om de verticale snelheid en de vlieghoogte aan te sturen, (3) botsingen ontwijken, wat neer komt op het bewaren van een veilige afstand tot gedetecteerde obstakels, en (4) navigatie, dat meerdere taken omhelst die relatie hebben met het nemen van beslissingen over (mogelijke) vliegrichtingen, en ook het bijhouden waar het toestel al geweest is.

Aanzienlijke vooruitgang is al geboekt in het realiseren van autonome taken op grotere soorten MAV's. Met kleine en lichte MAV's zijn er echter nog maar zeer beperkte mogelijkheden gedemonstreerd. Bij aanvang van deze studie was er nog geen oplossing gerealiseerd die zelfs maar de eerste drie niveaus van autonomie mogelijk maken op een kleine lichtgewicht MAV.

Dit onderzoek richt zich specifiek op de lichtgewicht klasse van MAV's met flappende vleugels. De natuur laat met veel voorbeelden zien dat dit principe van vliegen gebruikt kan worden op zeer kleine schaal. Studies hebben aangetoond dat sommige insectenvleugels drie maal zoveel opwaartse druk kunnen produceren dan men zou verwachten op basis van conventionele aerodynamische effecten. Bij het formaat van insecten is het aannemelijk dat aerodynamische krachten efficiënter

worden geproduceerd door flappende bewegingen dan roterende bewegingen. Een unieke eigenschap van flappende vleugels is dat zij een combinatie vormen van het genereren van opwaartse krachten, voortstuwende krachten en draaimomenten. Daarom vormen flappende vleugels een interessant concept voor het miniturariseren van MAV's. Hoewel de interesse in dit type MAV's groeit, is er nog weinig onderzoek gedaan naar het autonoom vliegen met deze toestellen. Het merendeel van de studies die daar wel naar zijn gedaan richten zich alleen op de eerste twee niveaus. Tot nu toe is er nog geen werk bekend waarin drie of zelf vier niveaus van autonome vlucht worden gecombineerd op een MAV met flappende vleugels. Om die reden is het onderzoeksdoel van deze studie als volgt geformuleerd:

> **Onderzoeksdoel**: Ontwikkel competenties die autonome vlucht mogelijk maken voor lichtgewicht MAV's met flappende vleugels.

Om dit doel te bereiken zijn er drie onderzoeksvragen opgesteld. Als startpunt van dit onderzoek is stereo visie geselecteerd als primaire sensor aan boord van het toestel. Verder is de DelFly geselecteerd als platform met flappende vleugels. De eerste onderzoeksvraag richt zich op deze combinatie van sensor en platform:

> **Onderzoeksvraag 1**: Hoe kan stereo visie worden gebruikt voor autonome vliegtaken op MAV's met flappende vleugels?

Een eerste stap in het beantwoorden van deze vraag is testen of stereo visie voldoende goed werkt op een MAV met flappende vleugels om obstakels te kunnen ontwijken, en of het beter werkt dan de meer gangbare methode waarbij een optisch stroomveld wordt gebruikt. De DelFly wordt uitgerust met een stereo visie systeem van 5,2 gram. Het verstuurt analoge videobeelden naar een grondstation die de beelden gebruikt in een stereo visie algoritme, en vervolgens stuurcommando's produceert die teruggestuurd worden naar het toestel. Een nieuwe methode om obstakels te ontwijken wordt geïntroduceerd dat bedoeld is om uitwijkmanoeuvres te genereren die altijd en in iedere situatie kunnen worden uitgevoerd zonder te botsen. Verschillende vluchten, met een vluchtduur tot aan 72 secondes, zijn uitgevoerd met dit systeem van 21 gram zonder dat het toestel botste. Deze resultaten demonstreren de grote potentie van het gebruik van stereo visie voor het ontwijken van obstakels, ook al maakt dit systeem gebruik van een processor die niet aan boord van het toestel zit.

Het volledig integreren van stereo visie aan boord van de DelFly is daarom gerealiseerd en getest als tweede stap. Een speciaal ontwikkeld stereo visie systeem van 4 gram wordt geïntroduceerd dat is uitgerust met alle functies die nodig zijn om het detecteren en ontwijken van obstakels mogelijk te maken. Een nieuwe uitvoering van het toestel wordt gepresenteerd, genaamd "DelFly Explorer". Het nieuwe ontwerp omvat een andere vorm van de staart en extra rolroeren, omdat uit eerdere tests bleek dat deze essentieel zijn. Vluchten met een lengte van meer dan negen minuten zijn behaald met dit systeem van 20 gram, enkel gelimiteerd door de capaciteit van de batterij. Een stereo visie algoritme genaamd "LongSeq" is geïntroduceerd dat geschikt is om live beelden te verwerken op dit systeem. Om beslissingen te nemen over uitwijkmanoeuvres wordt in deze tests gebruik gemaakt

van een reactief algoritme. Deze aanpak is getest in situaties met een beperkt aantal obstakels, waarbij voldoende tussenruimte is gelaten om uitwijkmanoeuvres mogelijk te maken.

De methode die in de eerste stap wordt geïntroduceerd om obstakels te ontwijken, wordt verder uitgebreid in de derde stap. Deze ontwijkstrategie, genaamd "Droplet", wordt toegepast op de DelFly Explorer. Deze methode blijkt beter geschikt voor het vliegen in complexe omgevingen, zoals ruimtes met veel obstakels dicht bij elkaar, en ruimtes afgeschermd door muren. Een ander stereo visie algoritme wordt toegepast dat tot meer robuust ontwijkgedrag leidt in combinatie met de Droplet-strategie. Afhankelijk van het type omgeving kunnen met deze aanpak vluchten van negen minuten worden gemaakt zonder te botsen.

Een alternatieve manier om de robuustheid en prestaties van het stereo visie systeem te vergroten is door te leren van eerdere observaties. Daarom wordt een tweede onderzoeksvraag voorgelegd:

> **Onderzoeksvraag 2**: Hoe kan een robot met één camera zichzelf leren om afstanden te schatten op basis van wat waargenomen wordt?

Een methode is ontwikkeld die leert hoe, op basis van het uiterlijk van objecten in de omgeving, de afstand tot die objecten kan worden geschat. Een systeem met één camera van 2 gram wordt hierbij gebruikt. In combinatie met lichte nabijheids sensoren om bijna-botsingen te detecteren, leert deze methode wat het uiterlijk is van de omgeving op het moment dat het toestel een object nadert. Omdat het geheugen van het camerasysteem slechts een klein aantal beelden kan opslaan, wordt een efficiënte methode voor beeldbeschrijving gebruikt om de video-data te comprimeren. Dit is essentieel voor het leerproces, omdat er data van voldoende verschillende plekken nodig zijn om te kunnen trainen. Een standaard aanpak gebruikmakend van k-NN blijkt alleen effectief wanneer grote hoeveelheden trainingsdata worden bewaard, wat niet mogelijk is met het camerasysteem dat gebruikt wordt in deze studie. Een tussenstap die de data clustert is daarom toegevoegd aan het trainingsproces, wat de hoeveelheid trainingsdata maximaliseert terwijl de variatie in data ook hoog blijft. Resultaten uit computersimulaties laten zien dat in beperkte ruimtes de leerprestaties slechts marginaal reduceren wanneer de clusterstap wordt toegepast. Deze simulaties laten ook zien dat door gebruik te maken van de geleerde informatie voor het schatten van afstanden, het toestel in staat is om bijna-botsingen te voorkomen. Echte testvluchten met het toestel van 19 gram laten zien dat de leercurve in de echte wereld minder steil is dan in de gesimuleerde wereld, maar dat dezelfde trends worden waargenomen. Door data op te slaan aan het einde van vluchten kan het toestel evengoed voldoende leren om vliegen zonder te botsen in een beperkte ruimte mogelijk te maken.

De eerste twee onderzoeksvragen focussen vooral op het gebruik van sensoren en algoritmes om autonome taken uit te voeren. Een andere aanpak zou echter kunnen zijn om het ontwerp van de robot aan te passen om diens autonome mogelijkheden te vergemakkelijken. Daarom is de derde onderzoeksvraag als volgt geformuleerd:

Onderzoeksvraag 3: Kunnen taken als het ontwijken van obstakels en andere navigatie-taken beter worden uitgevoerd als het vermogen voor aansturing van een MAV met flappende vleugels wordt verbeterd?

Een innovatief besturingsmechanisme wordt geïntroduceerd die vooral bedoeld is om de aansturing van de vliegrichting te verbeteren voor MAV's met flappende vleugels. Vooral wanneer obstakels te laat worden gedetecteerd om nog een normale uitwijkmanoeuvre uit te voeren, is het vermogen om stil te hangen en op de plaats om te draaien erg gunstig omdat dit een alternatieve en mogelijk meer veelzijdige en robuuste ontsnappingsweg biedt. Testvluchten laten inderdaad zien dat hoge draaisnelheden kunnen worden behaald en dat het toestel van 21 gram in staat is om de vliegrichting om te draaien terwijl het maar een zeer kleine beweegruimte nodig heeft. Het voorgestelde mechanisme maakt het ook mogelijk om de hellingshoek en de rolhoek aan te sturen. Hoewel er meer en zwaardere servo's gebruikt worden, maakt dit mechanisme het dus mogelijk om alle standhoeken aan te sturen zonder een staart nodig te hebben, wat voor dit toestel in een gewicht resulteert van net onder 20 gram.

Geconcludeerd kan worden dat een volledige implementatie van de eerste drie niveaus van autonome vlucht zijn gerealiseerd op een MAV met flappende vleugels: standaansturing, hoogte-aansturing en het ontwijken van obstakels. Voor standaansturing is een nieuw mechanisme voor vleugelaansturing gerealiseerd dat de vliegbegrenzingen verruimd met gecontroleerd stil hangen en ook met zijwaards vliegen. Dit verbetert de mogelijkheden voor het ontwijken van obstakels doordat het toestel gecontroleerd kan draaien in een heel klein gebied. Hoogte-aansturing op basis van barometrische drukmetingen is gerealiseerd, met redelijke prestaties tot gevolg. Aangetoond is dat stereo visie een haalbare en zeer effectieve oplossing is tot het ontwijken van obstakels. Het overtreft oplossingen die werken op het principe van een optisch stroomveld. Obstakels detecteren en ontwijken op basis van slechts één camera door middel van een leerproces blijkt haalbaar te zijn voor kleine ruimtes. Voor het ontwikkelen van de autonome taken in deze studie is gebleken dat het erg belangrijk is om rekening te houden met de relaties tussen sensoren, verwerkingssystemen, ontwerp van het toestel, het vlieggedrag van het toestel en de omgeving. Een experimentele aanpak is bovendien belangrijk om een goed inzicht te krijgen in al deze relaties.

Voor het vierde niveau van autonome vlucht, navigatie, zijn waardevolle inzichten opgedaan. Een potentiële navigatievaardigheid is om de hoeveelheid bezochte plekken te maximeren door repetitieve vliegpatronen te voorkomen. Het leren van het uiterlijk van de omgeving kan worden beschouwd als een effectieve manier om eerder bezochte plekken te herkennen. De nieuwe toegevoegde vliegeigenschappen, zoals stilhangen en zijwaarts vliegen, maken het uitvoeren van specifieke taken in nauwe omgevingen mogelijk, zoals vliegen door nauwe gangen en door geopende ramen en deuren. Dit maakt nieuwe manoeuvres en gedragingen mogelijk.

Aanbevelingen voor vervolgonderzoek kunnen verdeeld worden in drie hoofdrichtingen. De eerste richting omvat het integreren van het stereo visie systeem uit de eerste hoofdstukken op het toestel met het vleugelaansturingsmechanisme

uit hoofdtuk 6. De tweede richting omvat het ontwikkelen van kleinere versies van dezelfde MAV met flappende vleugels, door lichtere versies van de autopiloot en het stereo visie systeem te realiseren. De derde richting omvat het uitbreiden van de de navigatiemogelijkheden met meer geavanceerde methodes. Eerste stappen in dit proces kunnen worden gedaan op basis van inzichten in dit proefschrift. Het principe van het leren van het uiterlijk van de omgeving kan ook worden gebruikt om eerder bezochte ruimtes te herkennen. Recent ontwikkelde Time-of-Flight sensoren maken het vermoedelijk mogelijk om verscheidene autonome taken uit te voeren ten koste van slechts een kleine gewichtstoename. Deze zouden zeer nuttig kunnen zijn voor additionele taken zoals muren volgen, deuren detecteren, nauwe gangen doorkruisen, en afstanden schatten tot structuur-arme oppervlakken en transparante ramen. In combinatie met het huidige stereo visie systeem zouden deze sensoren het mogelijk moeten maken om autonoom te vliegen op alle vier de niveaus.

Contents

1 Introduction

Many types of flying robots have emerged over the last decade. In a few years time, these systems have turned into affordable machines that are used by governments, industries and private individuals. New applications in various sectors are announced on a daily basis: military operations, surveillance tasks, inspections, (agricultural) monitoring, filming and photographing, and even racing.

'Drones', as the general public calls them, have reached their popularity mainly due to the development of many small types. Making them smaller made them cheaper, more safe, and more user-friendly such that the number of possible users grew significantly. In scientific literature, small drones are called Micro Air Vehicles (MAVs). The term 'micro' covers a wide spectrum of possible sizes and masses, as visualized in Fig. 1.1. The maximum mass for an MAV is about a few kilograms. On the lower end of this spectrum, only time will tell what actually can be achieved in terms of miniaturization. Extremely small MAVs will play a significant role in indoor applications, as in such cluttered environments their small size is advantageous and often indispensable.

In the context of extremely small MAVs, the smallest MAV developed so far has a mass in the order of sub-grams [1]. Remarkably, this system is not a conventional multirotor, but a flapping wing vehicle (ornithopter). However, this is not surprising when one studies nature, which shows that flapping wing methods can be very effective at various scales of body size. For example, for certain types of insects it is determined that at low speeds the lift force produced by the insect wings is a factor two or three times higher than would be expected based on conventional aerodynamic effects [2]. Furthermore, the state-of-the-art of current actuation systems shows that, at insect scales, the aerodynamic forces are likely to be produced more efficiently by flapping motions (using piezoelectric muscles) than by rotational motions (using electric DC motors) [3]. Besides scalability, another advantage of the flapping wing concept is its unique solution of *combining* the generation of lift forces, thrust forces and control moments. For this reason

1

Figure 1.1: The weight spectrum of Micro Air Vehicles; from kilograms to sub-grams. The examples show the variety in types of platforms, and also indicate what types are common for each weight class. The 20 g DelFly Explorer, which is developed in this thesis, is indicated on the bottom row. **6.7 kg** Altura Zenith ATX8 [7]. **5 kg** AscTec Neo [8]. **2.9 kg** UX5 HP [9]. **2 kg** PIXHAWK [10]. **1.4 kg** Phantom 4 [11]. **750 g** DISCO FPV [12]. **650 g** AscTec Pelican [13]. **500 g** Bebop 2 [14]. **312 g** Robo Raven [15]. **130 g** Nano+ [16]. **62 g** Robotic Hummingbird [17]. **46 g** Ladybird V2 [18]. **20 g** DelFly Explorer (this thesis). **3 g** DelFly Micro [19]. **0.1 g** Robobee [1].

many insects have a very extensive flight envelope (hovering, flying forward, backward and sideways)[4], and many birds possess a remarkable agility [5], and can fly forward at high speeds with unsurpassed efficiency [6].

Many indoor applications will benefit from MAVs becoming fully autonomous. That will allow these vehicles to operate in areas that cannot be accessed by humans or other man-made vehicles for various reasons. Examples are spaces that are too narrow for humans to fit in, or situations that are dangerous or unhealthy. Furthermore, small MAVs could become extremely useful especially when they operate in groups or swarms. Autonomy is crucial in such cases as it allows operators to control whole swarms of MAVs at the same time, and autonomy can even make operators superfluous at some point [20].

The small and lightweight MAVs that are so promising still have very limited autonomous flight capabilities. The main reason for this is weight. Since MAVs need to lift their own weight, the sensing and processing devices that can be added is limited. The size, weight and power (SWaP) characteristics of such components all

influence the total required and total available power for flight. Small scale sensors are a necessity and developments in this field are important for the progress of MAV autonomy. Small sensors are typically the product of a compromise between SWaP requirements and quality [21]. To enable autonomous flight of fly-size (flapping wing) MAVs, it is therefore essential to not only select the best combination of sensors, but also to develop robust methods for using qualitatively poor sensors and little processing. Before delving into the proposed solution, first some of the state-of-the-art solutions to autonomous flight are discussed.

1.1. Autonomous capabilities of Micro Air Vehicles

The ability to perform autonomous flight will enable many new applications for MAVs. In particular, many indoor applications will become viable, as small-sized systems are beneficial in those situations. GPS-based navigation has already been introduced in many existing platforms, but this is not a complete solution. In indoor environments, and also in urban areas, GPS is often not reliable or available. Furthermore, in many of these cases, navigating a pre-planned route is not sufficient; while flying it is often necessary to avoid obstacles and to make decisions where to go when the environment is completely unknown [22].

In literature on MAV research, the term *navigation* is often used to cover all autonomous flight capabilities beyond the control of a vehicle's attitude and velocity. In the control systems community, however, navigation refers to determining the position, velocity and attitude of the vehicle; guidance is then defined as the determination of the desired path, and control as the execution of the guidance command while stabilizing the vehicle. In this thesis a different scheme is used that discerns autonomous flight capabilities using four levels [23]:

1. **Attitude Control**: controlling attitude angles; on some types of platforms necessary to enable stable flight, but also used to increase robustness against disturbances or to increase the agility of the platform.

2. **Height Control**: control of vertical speed and/or maintaining a certain distance to the ground (or ceiling), or maintaining a certain (barometric) altitude; combined with stable flight, if necessary by means of attitude control, this allows the vehicle to stay in the air.

3. **Collision Avoidance**: maintaining a safe distance between the vehicle and detected obstacles; combined with the two previous levels this allows a vehicle to remain in the air even when there are objects around. This may involve proactive path planning methods, but also reactive approaches. In the extreme case it can be a method to keep the vehicle in one place, in which case it is similar to height control but then in the horizontal plane.

4. **Navigation**: deciding where the vehicle needs to go; the previous levels will make the vehicle fly randomly through the environment or will make it hover in one place. Depending on the task, navigation may involve the selection of and guidance to waypoints, or making decisions of which direction to fly to.

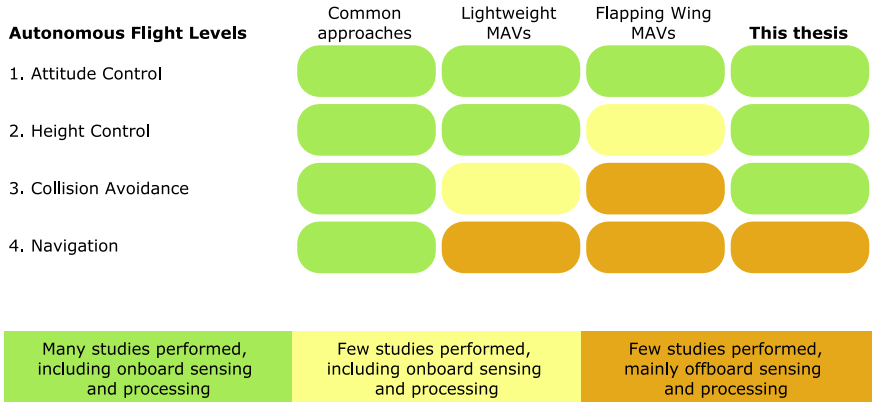

Figure 1.2: Overview indicating which levels of autonomous flight have been studied for different categories of MAVs. This thesis falls within the category of flapping wing MAVs.

The following sections give an overview of current approaches to autonomous flight of MAVs in the literature. First, it is discussed which approaches are common for MAVs in general and why most of these approaches are not suitable for lightweight vehicles. Second, it is discussed what autonomous capabilities on lightweight platforms have been realized, and what studies have been done on autonomous flight of flapping wing MAVs. Fig. 1.2 indicates for these different categories of MAVs which levels of flight autonomy have been studied. In this thesis the term *lightweight* is used for platforms under 50 g. The majority of flapping wing MAVs for indoor use indeed fall within this range.

1.1.1. Common approaches to autonomous flight

Considerable progress has been made in the development of autonomous flight systems for MAVs in general. Several types of sensors have been successfully demonstrated on board these platforms that provide information on the environment. Light Detection And Ranging (LIDAR) sensors are a form of laser scanning devices that measure distances to surrounding obstacles in a similar way as radar systems [24–30]. Furthermore, different types of vision sensors are used, such as RGB (color) [10, 16, 31–33] and infrared cameras, RGB-D cameras that not only provide color but also depth information per pixel [34–36], and event-based cameras that register pixel-level brightness changes with a very small delay [37–39]. Other commonly used sensors are acoustic sensors (e.g., ultrasound), mainly for altitude estimation but also for obstacle detection [40–42], and infrared ranging or time-of-flight sensors [43, 44]. Besides sensors that provide information about the environment (exteroception), Inertial Measurement Units (IMUs) have become an almost standard component of MAVs because they provide measurements on specific force and angular rates of the vehicle body (proprioception). IMUs often contain additional magnetometers for measuring the magnetic field around the body.

Autonomous flight can be achieved with a combination of sensors and sufficient

processing. However, some of the sensors are already too heavy by themselves to apply them to lightweight platforms. For example, LIDAR systems currently weigh more than 100 g. Cameras are typically much lighter, an important reason for their common use on MAVs, but for various vision tasks it is necessary to use high-quality (global shutter) cameras which have a mass typically above 10 g. The cameras need to be combined with sufficient processing power for extracting information from the images in real-time, which adds more weight and also demands a severe amount of power. Especially when fusing visual and inertial data [45–47], a process that provides highly accurate estimates of motion and attitude with a small degree of drift over time, the combination of a camera and a processor is typically too heavy to be applied to lightweight MAVs. Most studies use a combination of different types of sensors, resulting in a relatively heavy solution even when the individual sensors are lightweight.

Another reason why standard approaches are often not scalable is the computational load of the used algorithms. For collision avoidance and navigation a common and very suitable approach is to use a Simultaneous Localization And Mapping (SLAM) algorithm [10, 24–29, 31, 33, 34, 48]. SLAM algorithms combine (metric) point cloud data, mainly from LIDAR or vision systems, over time to construct a map from the environment. In doing so, the algorithm also allows to determine the position of the vehicle and keeps track of the path flown. The map is then used as a basis to perform obstacle detection and trajectory generation. However, these algorithms typically require a processing power in the order of 1 GHz or more [29]. Moreover, a lot of memory is required to store such maps, especially in the case of 3D maps, which can easily add up to 1 GB or more [27]. A second example of a common navigation method is visual or visual-inertial odometry [16, 45–47, 49]. Such algorithms keep track of a sparse set of image features to estimate the motion of the vehicle over time. The memory demands of these algorithms are much lower compared to SLAM, but as mentioned in the previous paragraph, these algorithms are computationally also very demanding. Besides, SLAM and odometry algorithms are not sufficient for autonomous flight, and require further processing steps, such as obstacle detection and collision avoidance, and trajectory planning [50].

1.1.2. Autonomous flight of lightweight Micro Air Vehicles

Due to the severe restrictions on available onboard processing power, studies on autonomous navigation of lightweight MAVs have demonstrated only very limited capabilities so far. A common approach for such platforms, mainly rotorcraft, is to use optical flow sensors that provide estimates of relative speed of a vehicle. In combination with other sensors that provide distance estimates, this makes it possible to estimate ground speed and vertical speed of the vehicle [18, 51–53]. Optical flow has also been used to avoid collisions and to fly along the centerline of a corridor [54–56]. To circumvent the issue of limited onboard processing power, some studies rely on off-board processing [57]. In other studies, onboard obstacle detection sensors are simply simulated by directly providing the locations of surrounding obstacles using an external positioning system [58, 59].

The majority of these studies implements optical flow-based velocity control and

obstacle avoidance on rotorcraft. The main reason for using optical flow sensors is that small and lightweight versions of these sensors can be made that provide low-resolution image data. This allows efficient visual data processing that does not require fast processing hardware. Another advantage of optical flow sensors is that they provide (scaleless) velocity measurements. Because rotorcraft have the property of drifting away over time, the optical flow inputs are necessary to control and stabilize the velocity of such platforms.

A downside of using optical flow sensors is their limited performance in detecting obstacles. The resolution of miniature versions of these sensors is relatively low, meaning that small objects are difficult to detect. Testing environments are often adapted such that they contain sufficient texture [55, 56]. Besides, the accuracy of optical flow measurements reduces with lowering the vehicle velocity, which is undesirable when flying indoors. Finally, the optical flow itself is small close to the Focus of Expansion, the image region that is in line with the direction of motion of the sensor. Measurement noise is therefore highest in this direction while it is also the most crucial direction in which obstacles should be detected [60].

In the light of this thesis, it is relevant to mention what studies have been performed on autonomous control of flapping wing MAVs. An important feature of flapping wing vehicles is that most designs include a tail that provides passive attitude stability. Besides, these vehicles also do not possess the drifting tendency of rotorcraft. Optical flow based velocity is therefore not a requisite to perform autonomous flight on these platforms. Nonetheless, optical flow sensors and monocular vision methods have been used to perform several flight tasks. Tracking of a visual target has been performed with onboard sensors and processing [61]; height control (on a 0.1 g platform) [62], obstacle detection [63], and line-following [64] were also demonstrated using onboard sensors. Besides, several studies have shown autonomous flight capabilities where a ground-based tracking system was used [1, 64–68].

In the field of flapping wing research, a major focus is put on attitude control of tailless platforms. This is deemed as a very challenging task since such platforms are not passively stabilized by a tail. Solving this task involves studying their mechanics, aerodynamics, materials, and flight controls. This focus is driven by inspiration from flying animals, which often show a wide variety of flight capabilities that provides them with broad flight envelopes. Due to the complexity of designing an active attitude control mechanism as part of a flapping wing mechanism, the number of studies focusing on higher levels of autonomous flight is limited to those listed in the previous paragraph.

Summarizing, very little work on autonomous flight levels 3 and 4, collision avoidance and navigation, has been done in the area of lightweight and flapping wing MAVs. No lightweight system has been demonstrated so far that can fully autonomously find its way through indoor environments, neither rotorcraft nor flapping wing vehicles. This thesis addresses these autonomous flight challenges as such capabilities will pave the way to many new useful applications of MAVS.

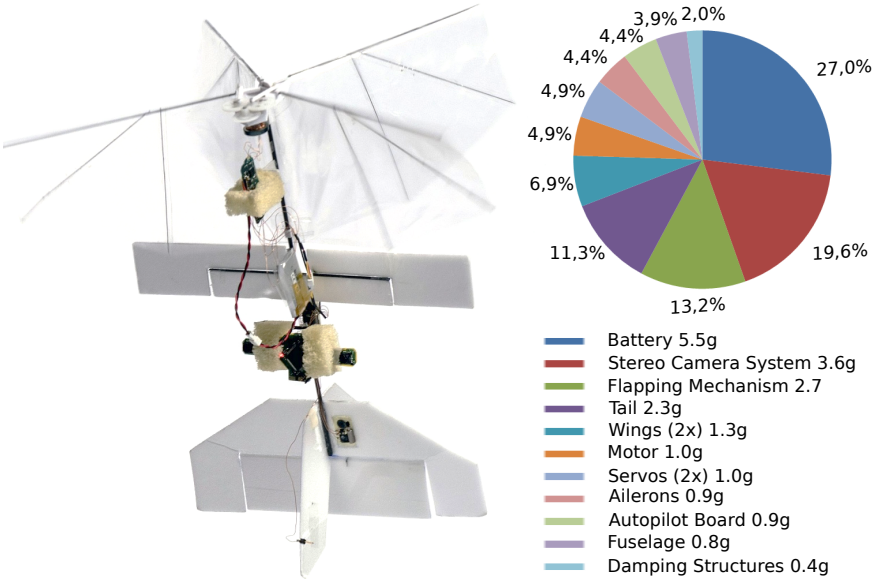

Figure 1.3: The 20 gram DelFly Explorer and a mass breakdown of its components. This version of the DelFly has a tail with a horizontal and vertical surface and an active elevator, and active aileron surfaces close to the wings.

1.2. Research Goal and Approach

No solution has been demonstrated so far that combines all levels of autonomous flight on a small and lightweight MAV. When looking specifically to flapping wing MAVs, the autonomous capabilities regarding levels 3 and 4 that have been demonstrated on these platforms is very limited. Therefore the following research goal is formulated.

Research Goal

Develop autonomous flight capabilities for lightweight flapping wing Micro Air Vehicles.

By focusing on flapping wing MAVs the autonomous flight solution being developed is not always applicable to other types of platforms. However, many solutions mentioned in the previous section are not applicable to flapping wing MAVs either. Because of the high potential of flapping wing MAVs for future indoor applications, it is necessary to explore what autonomous flight methods are suitable and effective on such platforms.

As a starting point of this research, stereo vision is selected as the primary sensor. Especially for detecting obstacles, this type of sensor is expected to outperform the more common monocular method of using optical flow. The combination of stereo vision and the flapping wing concept is unique in robotics, even though

Figure 1.4: Time lapse image showing the DelFly Explorer autonomously exploring a room by using its onboard stereo vision system for obstacle avoidance.

in nature it is very effectively used for depth perception by owls [69, 70]. The first research question is formulated as follows:

Research Question 1

How can stereo vision be used on flapping wing Micro Air Vehicles for autonomous flight tasks?

A preliminary study is first conducted in which a flapping wing MAV is equipped with a stereo vision system that relies on off-board processing. It is explored in this study which advantages and disadvantages stereo vision has, and how it can be used for autonomous navigation. The following step is to realize stereo vision on board a flapping wing MAV, as this makes the system fully autonomous. This poses challenges on different aspects of the system: creating a lightweight stereo vision system, developing a vision algorithm to detect obstacles, creating a flapping wing vehicle that can carry the sensor and that is able to follow its direction commands. The platform that is developed as a part of this study, the DelFly Explorer, is shown in Fig. 1.3. Furthermore, having the ability to detect obstacles does not guarantee collision-free flight as the onboard sensor has limitations and so does the platform. An avoidance strategy is therefore required that takes into account these limitations in order to effectively avoid collisions. Fig. 1.4 visualizes the end result of combining these steps: it shows the DelFly Explorer, equipped with a stereo vision system, flying around in a room fully autonomously.

Vision-based detection of obstacles is never fully reliable. On a lightweight MAV, severe restrictions on the resolution of the cameras and the amount of processing power are posed, which increases the limitations of such a system. To cope with these limitations it is useful to explore other methods for obstacle detection. Therefore, a method is developed that is based on previous work involving appearance-based recognition of earlier observed places. The second research question formulates an alternative method for obstacle detection using this approach:

Research Question 2

How can a monocular robot learn by itself to see distances to obstacles by means of appearance?

Both previous research questions have a main focus on using sensors and algorithms to perform autonomous flight tasks. However, by only focusing on these elements, a sub-optimal solution might be reached. For autonomous flight it is beneficial to have a broad flight envelope as this will allow more options for behaviors for the task of obstacle avoidance, and also navigation tasks. For example, being able to hover or even to fly backwards would allow for more options to avoid collisions with obstacles. Being able to perform fast transitions between different flight speeds has advantages when traversing narrow passages (e.g., doors) or when rejecting gust disturbances. Similar advantages can be thought of when lateral flight control is realized. By extending the capabilities of the vehicle, i.e., increasing the flight envelope, more combinations of flight modes and control strategies can be developed. This may lead to more optimal solutions for specific flight tasks, as the complexity of a certain task can be significantly reduced by using the right vehicle behavior. An analogy can be drawn with the task of writing with a pen. Some people are able to write using their mouth which is useful if one misses hands; however, people normally write with their hands because this makes the task much easier: they can see immediately what they write, and they can do it for long periods of time because they can sit upright.

The approach of adapting the robot's body to facilitate its autonomous flight tasks is inspired by the principle of *Embodied Intelligence*, which states that intelligent behavior emerges from the interplay between brain, body and world [71, 72]. In other words, the interplay between the payload, which serves as the brain of the MAV, and the design of the vehicle, its body, can be exploited to realize autonomous capabilities. This brings us to the third and final research question:

Research Question 3

Can the performance of the obstacle avoidance task and other navigation tasks be improved by increasing the control authority of flapping wing Micro Air Vehicles?

1.3. Research Scope

In order to accomplish the research goal of this thesis, the research is subject to several limitations and assumptions.

The vehicle that has been selected for this research is the DelFly [23]. Flapping wing vehicles are not yet very common, and are therefore not widely available commercially. The DelFly has been developed within the research group and has proven to be a reliable system with a good performance. It is able to perform long flights of over fifteen minutes and can carry a decent payload. This provides

the possibility to carry various sensors on board and to test their effectiveness in the loop. By using the existing DelFly design as a starting point, experimental data can be gathered at an early stage of this research. Based on test results and new insights the design of the vehicle is further developed or changed during this research.

Stereo vision cameras are selected as the main sensors for obtaining information about the environment. Cameras are regarded to be information-dense; they produce a lot of visual information for a relatively low weight and low power consumption. An important focus of this research is to show that a stereo setup is beneficial over a monocular setup. Other types of (non-vision) sensors are initially ignored because of their lower information-density. It is assumed that the light conditions in the environment are always suitable for the used camera. Test locations are selected as such. Taking care of poor light conditions, either by using different visual sensors or by using active illumination, is regarded to be beyond the scope of this research. Besides sensors for observing the environment, also sensors for observing the state of the vehicle are used. These are typically small chip-based sensors that are already integrated within the autopilot board of the DelFly.

This research focuses on indoor autonomous flight. Therefore calm wind conditions are assumed. This means that there would be no air flow through the environment. In practice, this is not realistic for many environments. Climate control systems form a common source for wind disturbances. Because a good disturbance rejection performance was not realized on the DelFly platform at the start of this research, wind disturbances are ignored and mostly prevented during tests. At the same time, improved attitude estimation and control are addressed to work towards a platform design that is better suited to cope with wind disturbances.

1.4. Research Contributions

The main contributions following from this thesis are listed as follows:

- The first study is presented where a flapping wing MAV is equipped with onboard stereo vision sensing. Flight test results are obtained verifying that this approach has significant advantages compared to a more traditional approach that uses optical flow. This is illustrated by flight experiments showing collision-free flights of over 1 minute.
- A compact stereo vision system of 4 grams is presented which is suitable for use on lightweight MAVs, such as the flapping wing vehicle which is a part of this thesis. The system includes a processor that runs efficient stereo vision algorithms to enable real-time obstacle detection.
- The DelFly Explorer is presented, the lightest MAV presented so far that can fly fully autonomous by maintaining a safe height above the ground an by avoiding obstacles. The total mass of the vehicle including stereo vision system is 20 grams. Results of autonomous flights are shown where collision-free flight is realized for as long as the battery lasts (up to 9 minutes), indicating that a very robust performance is obtained.

- A computationally efficient obstacle avoidance strategy is introduced that relies on stereo vision information for ensuring collision-free flight. The method is suitable for flying in narrow and cluttered environments and provides theoretically guaranteed safety. By taking into account nonholonomic constraints of the flapping wing vehicle in this study and limitations of the used stereo vision system, the performance of the method in combination with the real platform is very reliable. This is validated by simulation experiments and real world experiments with the DelFly Explorer.

- A self-supervised learning method is proposed and tested for performing obstacle detection. The method allows the use of monocular vision without requiring optical flow processing. It is demonstrated that the frequency of near-collisions decreases considerably with learning time.

- A new wing control mechanism for attitude control on flapping wing MAVs is presented. The mechanism is combined with a 1 gram autopilot that performs onboard attitude estimation. This combination enables the vehicle to perform fully autonomous attitude control without requiring a tail. Furthermore, results are presented showing that in combination with a tail, this mechanism provides an extensive flight envelope with the following capabilities: hovering, flying forward, backward and sideways, fast transitioning between flight regimes, and agile maneuvering. It is shown that these capabilities lead to more robustness in performing obstacle avoidance.

1.5. Outline of the Thesis

The organization of the thesis is described in this section, with a visual outline shown in Fig. 1.5. All chapters are based on either conference or journal publications. These publications are therefore included "as is". As a result, the introduction sections of the chapters contain a certain amount of overlap. For this reason each chapter is preceded by a short introduction that briefly explains how each chapter fits within the scope of this thesis.

Chapter 2 explores the applicability of stereo vision on board the DelFly flapping wing MAV for the purpose of obstacle avoidance. The stereo vision images are processed off-board, which makes it possible to test different types of existing stereo vision algorithms. The study gives insight into how stereo vision should be applied for obstacle avoidance and allows to compare the effectiveness of several reactive avoidance strategies.

- Chapter 2 is based on the following book chapter:
  S. Tijmons, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, H.M. Ruijsink, E. van Kampen, Q.P. Chu, *Stereo Vision Based Obstacle Avoidance on Flapping Wing MAV's*, Advances in Aerospace Guidance, Navigation and Control, (2013).

Based on these insights, Chapter 3 describes a new design for the flapping wing MAV, which is named the DelFly Explorer. It includes an innovative lightweight

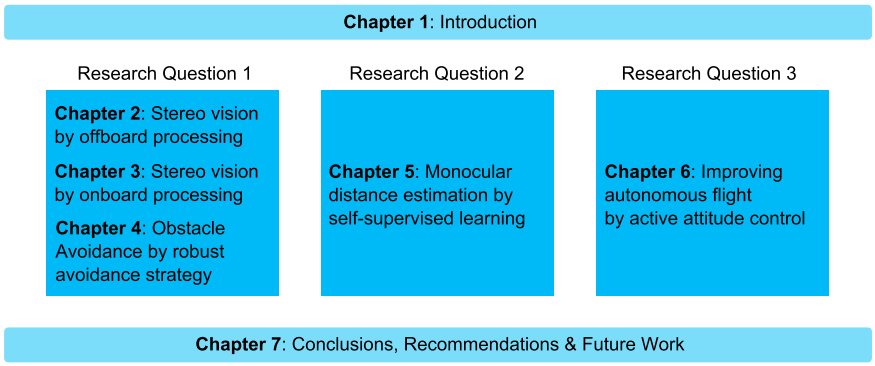

Figure 1.5: Visual outline of the thesis.

stereo vision camera system that enables onboard sensing and processing. Initial flight test results are presented, as well as results on an efficient stereo vision algorithm that runs in real time on the system.

- Chapter 3 is based on the following conference publication:
  C. De Wagter, **S. Tijmons**, B.D.W. Remes, G.C.H.E. de Croon, *Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System*, In: IEEE International Conference on Robotics and Automation (ICRA), (2014).

An efficient and robust obstacle avoidance algorithm is presented in Chapter 4. The algorithm is specifically suitable for using on board the flapping wing MAV design as presented in Chapter 3. A different stereo vision algorithm is proposed and implemented, and a thorough analysis of the proposed avoidance strategy is presented. Its effectiveness and robustness is demonstrated by both computer simulations and real flight experiments.

- Chapter 4 is based on the following journal publication:
  S. Tijmons, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, M. Mulder, *Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV*, In: IEEE Transactions on Robotics, (2017).

Chapter 5 explores the applicability of appearance-based learning for the task of estimating distances to obstacles. Flight tests are performed to demonstrate the effectiveness of this method.

- Chapter 5 is based on the following conference publication:
  K. Lamers, **S. Tijmons**, C. De Wagter, G.C.H.E. de Croon, *Self-supervised monocular distance learning on a lightweight micro air vehicle*, In: International Conference on Intelligent Robots and Systems (IROS), (2016).

A new concept for using wing-based control is introduced in Chapter 6 which enables more agile flight maneuvers. The effectiveness of the proposed control

concept is first analyzed by static experiments. In addition, real flight experiments demonstrate that the vehicle can perform fast turn-around maneuvers using this control concept. Such maneuvers increase the robustness of the system in performing obstacle avoidance by allowing near-collision avoidance. Finally experiments are performed, which show that the presented control concept enables the vehicle to perform tailless hover flight.

- Chapter 6 is based on the following journal publication:
  S. Tijmons, M. Karásek, G.C.H.E. de Croon, *Attitude control system for a lightweight flapping wing MAV*, In: Bioinspiration & Biomimetics (SUBMITTED),

  which is an extension of the following conference publication:
  J.L Verboom, **S. Tijmons**, C. De Wagter, B.D.W. Remes, R. Babuska, G.C.H.E. de Croon, *Attitude and altitude estimation and control on board a Flapping Wing Micro Air Vehicle*, In: IEEE International Conference on Robotics and Automation (ICRA), (2015).

Finally, Chapter 7 presents the final conclusions and recommendations for future research.

References

[1] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, *Controlled Flight of a Biologically Inspired, Insect-Scale Robot.* Science **340**, 603 (2013).

[2] A. P. Willmott and C. P. Ellington, *The mechanics of flight in the hawkmoth manduca sexta. i. kinematics of hovering and forward flight.* Journal of Experimental Biology **200**, 2705 (1997).

[3] E. E. Steltz, *Redesign of the micromechanical flying insect in a power density context* (ProQuest, 2008).

[4] N. Sapir and R. Dudley, *Backward flight in hummingbirds employs unique kinematic adjustments and entails low metabolic cost,* Journal of Experimental Biology **215**, 3603 (2012).

[5] A. L. Thomas, *The flight of birds that have wings and a tail: variable geometry expands the envelope of flight performance,* Journal of Theoretical Biology **183**, 237 (1996).

[6] D. Lentink, U. Müller, E. Stamhuis, R. De Kat, W. Van Gestel, L. Veldhuis, P. Henningsson, A. Hedenström, J. J. Videler, and J. L. Van Leeuwen, *How swifts control their glide performance with morphing wings,* Nature **446**, 1082 (2007).

[7] *Altura Zenith ATX8,* https://www.aerialtronics.com/altura-zenith-engels/altura-zenith (2017).

[8] *AscTec Neo,* http://www.asctec.de/en/uav-uas-drones-rpas-roav/asctec-neo/ (2017).

[9] *Trimble UX5 HP,* http://uas.trimble.com/ux5-hp (2017).

[10] K. Schauwecker and A. Zell, *On-board dual-stereo-vision for the navigation of an autonomous MAV,* Journal of Intelligent & Robotic Systems **74**, 1 (2014).

[11] *DJI Phantom 4,* https://www.dji.com/phantom-4 (2017).

[12] *Parrot DISCO FPV,* https://www.parrot.com/us/drones/parrot-disco-fpv#app (2017).

[13] *AscTec Pelican,* http://www.asctec.de/uav-uas-drohnen-flugsysteme/asctec-pelican/ (2017).

[14] *Parrot Bebop 2,* https://www.parrot.com/us/Drones/Parrot-bebop-2 (2017).

[15] L. Roberts, H. A. Bruck, and S. K. Gupta, *Autonomous Loitering Control for a Flapping Wing Miniature Aerial Vehicle with Independent Wing Control,* in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2014) pp. V05AT08A013–V05AT08A013.

[16] C. Forster, M. Pizzoli, and D. Scaramuzza, *SVO: Fast semi-direct monocular visual odometry,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 15–22.

[17] D. Coleman, M. Benedict, V. Hrishikeshavan, and I. Chopra, *Design, development and flight-testing of a robotic hummingbird,* in *AHS 71st Annual Forum, Virginia Beach, Virginia, edited by AH Society* (2015).

[18] A. Briod, J.-C. Zufferey, and D. Floreano, *Optic-flow based control of a 46g quadrotor,* in *Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments, IROS 2013*, EPFL-CONF-189879 (2013).

[19] *DelFly Micro,* http://www.delfly.nl/micro/ (2017).

[20] A. E. Ortiz and C. Langbort, *On multi-UAV scheduling for human operator target identification,* in *American Control Conference (ACC), 2011* (IEEE, 2011) pp. 1837–1842.

[21] M. Perlmutter and L. Robin, *High-performance, low cost inertial MEMS: A market in motion!* in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION* (IEEE, 2012) pp. 225–229.

[22] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, *et al.*, *Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments,* IEEE Robotics & Automation Magazine **21**, 26 (2014).

[23] G. C. H. E. de Croon, M. Percin, B. D. W. Remes, R. Ruijsink, and C. De Wagter, *The DelFly: Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot* (Springer, 2015).

[24] S. Grzonka, G. Grisetti, and W. Burgard, *A fully autonomous indoor quadrotor,* IEEE Transactions on Robotics **28**, 90 (2012).

[25] I. Dryanovski, R. G. Valenti, and J. Xiao, *An open-source navigation system for micro aerial vehicles,* Autonomous Robots **34**, 177 (2013).

[26] S. Shen, N. Michael, and V. Kumar, *Autonomous multi-floor indoor navigation with a computationally constrained MAV,* in *Robotics and automation (ICRA), 2011 IEEE international conference on* (IEEE, 2011) pp. 20–25.

[27] A. Bachrach, S. Prentice, R. He, and N. Roy, *RANGE–Robust autonomous navigation in GPS-denied environments,* Journal of Field Robotics **28**, 644 (2011).

[28] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stückler, and S. Behnke, *Multilayered mapping and navigation for autonomous micro aerial vehicles,* Journal of Field Robotics (2015).

[29] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke, *Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments,* Journal of Intelligent & Robotic Systems **1**, 1 (2015).

[30] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, *River mapping from a flying robot: state estimation, river detection, and obstacle mapping,* Autonomous Robots **33**, 189 (2012).

[31] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, *Vision-based autonomous mapping and exploration using a quadrotor MAV,* in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (IEEE, 2012) pp. 4557–4564.

[32] A. D. Wu, E. N. Johnson, M. Kaess, F. Dellaert, and G. Chowdhary, *Autonomous flight in GPS-denied environments using monocular vision and inertial sensors,* Journal of Aerospace Information Systems (2013).

[33] S. Weiss, D. Scaramuzza, and R. Siegwart, *Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments,* Journal of Field Robotics **28**, 854 (2011).

[34] G. Flores, S. Zhou, R. Lozano, and P. Castillo, *A vision and GPS-based real-time trajectory planning for a MAV in unknown and low-sunlight environments,* Journal of Intelligent & Robotic Systems **74**, 59 (2014).

[35] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, *Visual odometry and mapping for autonomous flight using an RGB-D camera,* in *Robotics Research* (Springer, 2017) pp. 235–252.

[36] R. G. Valenti, I. Dryanovski, C. Jaramillo, D. P. Ström, and J. Xiao, *Autonomous quadrotor flight using onboard RGB-D visual odometry,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 5233–5238.

[37] A. Censi, J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza, *Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor,* in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (IEEE, 2013) pp. 891–898.

[38] E. Mueggler, B. Huber, and D. Scaramuzza, *Event-based, 6-DOF pose tracking for high-speed maneuvers,* in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (IEEE, 2014) pp. 2761–2768.

[39] A. Censi and D. Scaramuzza, *Low-latency event-based visual odometry,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 703–710.

[40] M. Becker, R. C. B. Sampaio, S. Bouabdallah, V. Perrot, and R. Siegwart, *In flight collision avoidance for a Mini-UAV robot based on onboard sensors,* Journal of the Brazilian Society of Mechanical Sciences and Engineering **2**, 12 (2012).

[41] J. Müller, A. V. Ruiz, and I. Wieser, *Safe & sound: A robust collision avoidance layer for aerial robots based on acoustic sensors,* in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION* (IEEE, 2014) pp. 1197–1202.

[42] A. Viquerat, L. Blackhall, A. Reid, S. Sukkarieh, and G. Brooker, *Reactive collision avoidance for unmanned aerial vehicles using doppler radar,* in *Field and Service Robotics* (Springer, 2008) pp. 245–254.

[43] S. Lai, K. Wang, K. Li, and B. M. Chen, *Path planning of rotorcrafts in unknown environment,* in *Control Conference (CCC), 2016 35th Chinese* (IEEE, 2016) pp. 10900–10905.

[44] K. Li, K. Zhang, and B. M. Chen, *On-board visual odometry and autonomous control of a quadrotor micro aerial vehicle,* in *Control and Automation (ICCA), 2016 12th IEEE International Conference on* (IEEE, 2016) pp. 68–73.

[45] J. Engel, J. Sturm, and D. Cremers, *Camera-based navigation of a low-cost quadrocopter,* in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (IEEE, 2012) pp. 2815–2821.

[46] V. Usenko, J. Engel, J. Stückler, and D. Cremers, *Direct visual-inertial odometry with stereo cameras,* in *Robotics and Automation (ICRA), 2016 IEEE International Conference on* (IEEE, 2016) pp. 1885–1892.

[47] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, *IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,* (Georgia Institute of Technology, 2015).

[48] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-scale direct monocular SLAM,* in *European Conference on Computer Vision* (Springer, 2014) pp. 834–849.

[49] M. Pizzoli, C. Forster, and D. Scaramuzza, *REMODE: Probabilistic, monocular dense reconstruction in real time,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 2609–2616.

[50] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, *Stereo vision-based obstacle avoidance for micro air vehicles using disparity space,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3242–3249.

[51] G. Barrows, T. Young, C. Neely, A. Leonard, and S. Humbert, *Vision Based Hover in Place,* in *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (2006) p. 586.

[52] J.-C. Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano, *A 10-gram vision-based flying robot,* Advanced Robotics **21**, 1671 (2007).

[53] A. Briod, J.-C. Zufferey, and D. Floreano, *A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments,* Autonomous Robots **40**, 789 (2016).

[54] J.-C. Zufferey, A. Beyeler, and D. Floreano, *Near-obstacle flight with small UAVs,* in *UAV" 2008*, LIS-CONF-2008-020 (Springer Verlag, 2008).

[55] J.-C. Zufferey, A. Beyeler, and D. Floreano, *Optic flow to steer and avoid collisions in 3D,* in *Flying Insects and Robots* (Springer, 2009) pp. 73–86.

[56] R. J. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, *Autonomous MAV guidance with a lightweight omnidirectional vision sensor,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3856–3861.

[57] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, *Visual-inertial navigation for a camera-equipped 25g nano-quadrotor,* in *IROS2014 aerial open source robotics workshop* (2014) p. 2.

[58] M. Pivtoraiko, D. Mellinger, and V. Kumar, *Incremental micro-UAV motion replanning for exploring unknown environments,* in *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (IEEE, 2013) pp. 2452–2458.

[59] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, *Motion primitives and 3D path planning for fast flight through a forest,* The International Journal of Robotics Research **34**, 357 (2015).

[60] A. Beyeler, J.-C. Zufferey, and D. Floreano, *Vision-based control of near-obstacle flight,* Autonomous robots **27**, 201 (2009).

[61] S. S. Baek, F. G. Bermudez, and R. Fearing, *Flight control for target seeking by 13 gram ornithopter,* in *IEEE/RSJ Int Conf on Intelligent Robots and Systems.* (2011).

[62] P. E. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, *Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor.* in *ICRA* (2012) pp. 4228–4235.

[63] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[64] G. C. H. E. de Croon, K. M. E. de Clerq, R. Ruijsink, B. D. W. Remes, and C. de Wagter, *Design, aerodynamics, and vision-based control of the DelFly,* International Journal on Micro Air Vehicles **1**, 71 (2009).

[65] S. H. Lin, F. Y. Hsiao, C. L. Chen, and J. F. Shen, *Altitude control of flapping-wing MAV using vision-based navigation,* in *American Control Conference (ACC), 2010* (IEEE, 2010) pp. 21–26.

[66] F. Y. Hsiao, H. K. Hsu, C. L. Chen, L. J. Yang, and J. F. Shen, *Using stereo vision to acquire the flight information of flapping-wing MAVs,* Journal of Applied Science and Engineering **15**, 213 (2012).

[67] S. S. Baek and R. S. Fearing, *Flight forces and altitude regulation of 12 gram I-Bird,* in *IEEE RAS and EMBS Int Conf on Biomedical Robotics and Biomechatronics (BioRob)* (2010) pp. 454–460.

[68] R. C. Julian, C. J. Rose, H. Hu, and R. S. Fearing, *Cooperative control and modeling for narrow passage traversal with an ornithopter MAV and lightweight ground station,* in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2013) pp. 103–110.

[69] R. F. van der Willigen, B. J. Frost, and H. Wagner, *Stereoscopic depth perception in the owl,* Neuroreport **9**, 1233 (1998).

[70] A. N. Iwaniuk and D. R. Wylie, *The evolution of stereopsis and the Wulst in caprimulgiform birds: a comparative analysis,* Journal of Comparative Physiology A **192**, 1313 (2006).

[71] F. J. Varela, E. Rosch, and E. Thompson, *The Embodied Mind: Cognitive Science and Human Experience* (MIT press, 1991).

[72] A. Clark, *Being there: Putting brain, body, and world together again* (MIT press, 1998).

2

Stereo Vision based Obstacle Avoidance on a Flapping Wing MAV with Off-board Vision Processing

Obstacle avoidance is a major element of the autonomous navigation task for MAVs. As was mentioned in the Introduction, a common approach to perform this task is to use a single camera and to perform optical flow calculations on the images. Previous studies have shown that optical flow methods suffer from vehicle vibrations when the camera is mounted on a flapping wing MAV. Furthermore, these methods provide poor information about obstacles present in the direction of motion.

This chapter explores whether stereo vision can be used as an alternative for optical flow. The aim of this study is to verify whether images from a stereo vision system on board a real flapping wing MAV provide sufficiently accurate information on the presence of obstacles. A small stereo vision camera system is mounted on board the DelFly II. The camera images are processed off-board. Several stereo vision algorithms from the literature are compared, and the performance of the system is analyzed. Three obstacle avoidance strategies are tested on this system to compare their effectiveness and robustness.

This chapter is based on the following article:
S. Tijmons, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, H.M. Ruijsink, E. van Kampen, Q.P. Chu, *Stereo Vision Based Obstacle Avoidance on Flapping Wing MAV's*, Advances in Aerospace Guidance, Navigation and Control, (2013)

2.1. Introduction

Autonomous flight of flapping wing MAVs (FWMAVs) is a considerable challenge. The main reason for this is that their lightweight prevents the use of heavy and energy-consuming laser scanners that are successful on heavier MAVs such as quadrotors [1] [2]. Still, there have been several attempts at achieving autonomous flight with FWMAVs. Hines et al. [3] describes a FWMAV design that is currently not able to fly on its own, but experiments show it is able to control its pitch and roll angle by using actuators that change the wing shape kinematics. Lin et al. [4] shows the altitude control of the 10 gram FWMAV called *Golden Snitch*. No onboard processing or sensing is used for this task. Using an external stereo camera the position of the vehicle is determined, and further control is performed by a ground station. Duhamel et al. [5] presents an experiment with a 101 milligram flapping wing microrobot called *RoboBee*. Using an onboard optical flow sensor and a well textured screen, the altitude is successfully controlled off-board in a closed-loop experiment, with only small oscillations and a slight drift. Baek et al. [6] performs closed-loop altitude control on a 12 gram ornithopter by using an external camera. In a follow up on this research [7], a 13 gram ornithopter is presented that is able to fly autonomously to a target, using an onboard infrared sensor for target tracking and 3-axis gyroscopes for attitude estimation. During 20 trials a success rate of 85% is reached. Garcia Bermudez et al. [8] performes optical flow measurements on a 7 gram ornithopter. Heavily down-sampled onboard camera images are stored on board during flight, and uploaded to a computer afterwards to compute optical flow. The main finding is a strong coupling between body motion and the sensed optical flow. Tedrake et al.[9] shows autonomous flight of an ornithopter with a 2-meter wingspan. Only pitch control has been tested successfully using an IMU.

With DelFly II several autonomy experiments have been performed dealing with various control tasks [10]. These tests range from height control with an external camera to height control and path following with an onboard camera and off-board processing. Also a novel appearance cue for obstacle avoidance is introduced [11] [12]. It is based on the principle that when an object is approached, its colors and detailed texture become more and more visible, while other objects move out of sight. It is shown that this cue is a useful complement to optical flow for detecting obstacles with the DelFly.

This experiment showed that optical flow is still not sufficient to perform obstacle avoidance on FWMAVs. To perform good optical flow measurements the camera images should be noiseless and rotation rates should be known, requiring three gyroscopes that can measure the rotational speeds of the vehicle. Measurements should be performed on board, but the amount of onboard processing power is currently too limited. Therefore the video signal is sent to a ground station, which implies a low frame rate. The frame rate of 30 FPS and line-by-line recording of the camera result in large image distortions that affect the optical flow quality.

In this thesis the use of stereo vision is proposed to circumvent these problems. Optical flow relies on image sequences, while stereo vision uses images taken at the same time. Vehicle motion has therefore a smaller influence on the quality of the measurements and the video frame rate is of no importance on the quality

of individual measurements. Furthermore, it gives an instantaneous overview of obstacles in sight of the camera.

In Section 2.2, a description is given of the DelFly system including stereo cameras and ground station. Section 2.3 discusses stereo vision and the algorithm used in this study. The performance of the stereo vision system is presented in Section 2.4. Closed-loop autonomy experiments and their results are discussed in Section 2.5. Finally a summary of the conclusions is given in Section 2.6.

2.2. System Design

2.2.1. Platform design

Since the research in this study focused on FWMAVs, tests were performed with the DelFly II. Its design is shown in Fig. 2.1. The most defining feature of the DelFly is that there is always a camera and transmitter on board (in this study two cameras). The current version of DelFly II is also equipped with gyrometers, a pressure meter, and onboard processing for these high-frequency measurements. Additional defining features are its biplane wing model and its tail. For more details, the interested reader is referred to [12]. Fig. 2.2 shows an overview of all system components and their interactions.

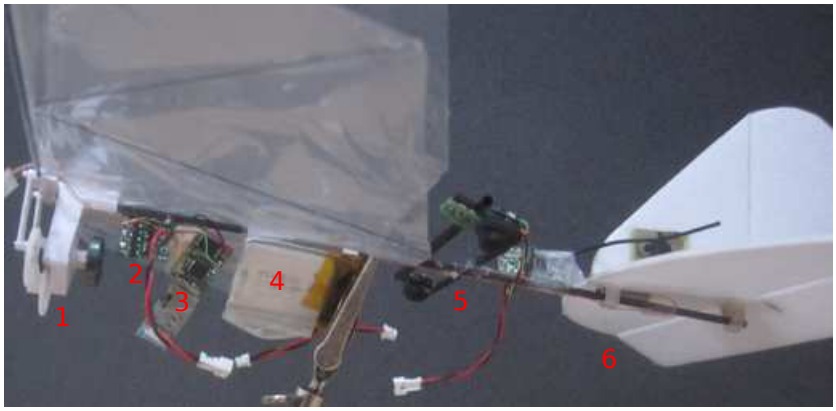

Figure 2.1: Side-view of DelFy II including stereo vision cameras. **1**: flapping mechanism. **2**: electronic speed controller. **3**: autopilot board. **4**: battery. **5**: stereo vision system. **6**: tail with servo-actuated rudder and elevator

For communication with the ground station, a Bluetooth transceiver is used. This system operates at the same frequency as the NTSC transmitter of the stereo system: 2.4GHz. Wi-Fi networks normally operate around this frequency as well. As a result, the images received on the ground can become noisy, as is illustrated in Fig. 2.3. The ground station uses a 2.30GHz dual-core system running on Windows 7. The system is prone to several types of delay. It takes around 60ms to receive the stereo images on the ground. Processing is then performed in real-time (40 ms) and control signals are then sent via Bluetooth. This is the slowest step, which at least takes around 60ms. However, because of interference from the other systems

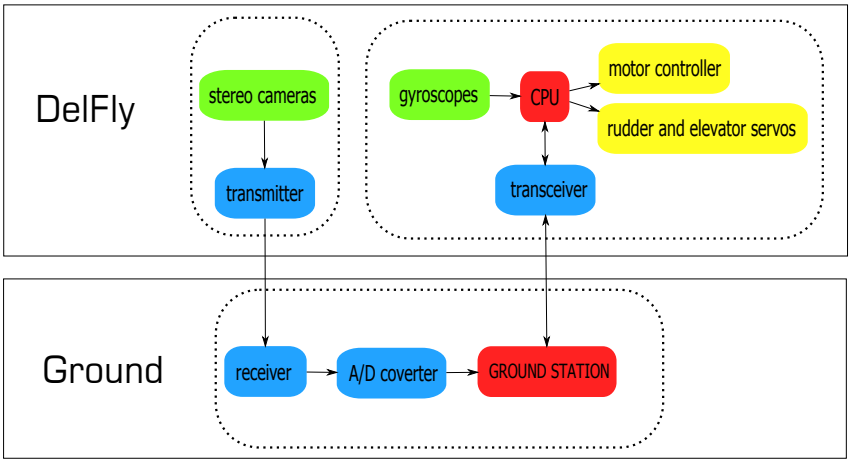

Figure 2.2: Diagram of the interaction between all system components

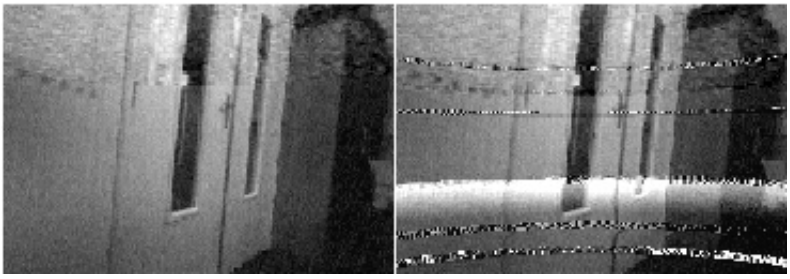

Figure 2.3: Example of noise due to an interfering source. Left is noise free, right contains severe noise

operating around the same frequency, this delay varies over time and can become more than 200ms in some cases.

The main feature of interest is the stereo vision camera, which will be discussed in more detail. Due to the stereo camera system, the weight of the DelFly in the current configuration is heavier than usual. Normally the total weight including sensors and batteries is under 17 gram. However, the stereo vision system, including a separate battery, accounts for 5.2 gram. The total weight of the DelFly in this configuration is 21.1 gram.

The selected configuration of the DelFly for this study is for slow forward flight because of the purpose of indoor obstacle avoidance. In this configuration, the speed can still be increased to several meters per second, but it can also fly stable with only 0.6m/s. Hovering is not possible due to the heavy weight of the configuration. The speed is controlled by the tail elevator. The rudder can be used to make turns. The turn speed can be controlled accurately with a servo. However, there is variation in the response of the DelFly to a rudder input. The turns are therefore not strictly circular. Furthermore, giving too much rudder input will result

in a fast spiral motion. Still, with sufficient rudder input, the turn diameter is less than 1m.

2.2.2. stereo vision camera system design

The stereo camera system is the main sensor of the DelFly in this study. Its components can be seen in more in detail in Fig. 2.4. The setup consists of two synchronized CMOS 720x240 cameras (with an offset of 7.6 cm) running at 25 Hz and a 2.4 GHz NTSC transmitter. The cameras have a field of view of ± 60 degrees horizontally. Because there is only one transmitter, the video streams from both cameras have to be combined as one. In the initial setup, this was done as follows: an NTSC frame consist of an even field and an odd field. To combine two synchronized NTSC cameras, the even lines of the first camera are scanned first, and then the camera source is switched and the uneven lines of the second camera are scanned. This image-based scheme results in frames which consist of image lines from the left and right camera alternately. The resulting frame size is still the same (720x480) but the resolution for each camera has now been reduced to 720x240 pixels.

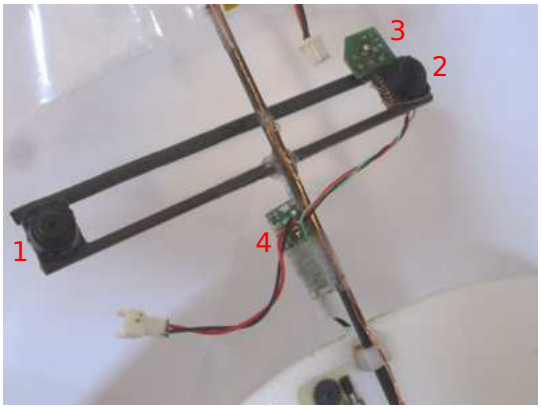

Figure 2.4: Stereo camera system. The base line of the cameras is 7.6cm. **1**: right camera. **2**: left camera. **3**: video stream merging device. **4**: NTSC transmitter.

During early tests with the camera system a shortcoming of this setup was noticed. The result from the stereo matching process was strongly affected by the motion of the camera. During static tests the results were promising and proved to be reliable, but during motion the results would become distorted. Since all even lines are scanned before the uneven lines in this image-based scheme, there is a time difference between the scan lines from the left and the right image. The first line of the 'transmitted' image comes from the right camera, the second line comes from the left camera. When the camera is at rest, it can be roughly assumed that these two lines are observing the same features. When the camera is in motion, this assumption does not hold anymore because of the time difference of approximately 20ms (half the time between two frames) between the lines. During this time the cameras might have changed orientation and the left and right image lines cover

different view directions. As a result, the output from the stereo matching process becomes distorted.

The hardware of the camera system was changed such that each time after a scan line has been scanned, the system switches to the other camera. As a result of this line-based scheme the frames sent by the transmitter now consist of two sets of two images that have been taken at different times. This is illustrated in Fig. 2.5. Two images (one from the left camera and one from the right camera) are captured on the even lines first (light colors), and after that another set of stereo images is captured on the uneven lines (dark colors). The images on the uneven lines are always the most recent stereo images, and these are used for stereo processing. Each individual image now has a resolution of 720x120 pixels. The benefit of this approach is that the time difference between the stereo images has been reduced significantly. Instead of switching between cameras after 240 lines have been scanned, switching is now done after each single scanline.

So by changing the hardware synchronization from an image-based scheme to a line-based scheme, the time difference has been reduced with a factor 1/240 to roughly 83μs. For the purpose of stereo matching it is assumed that two consecutive uneven image lines (which always contain image lines from both cameras) cover the same image areas.

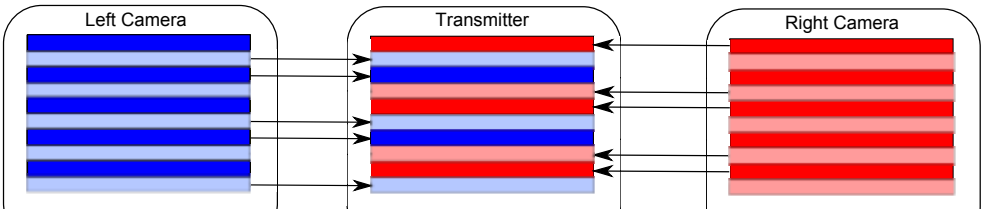

Figure 2.5: Line-based synchronization scheme designed for FWMAV stereo vision. The 'transmitted' image consists of image lines from the left and right camera's. The even lines (light) are scanned first and consist of image lines from the left (blue) and right (red) camera alternately. It takes about 83μs to scan one image line. After all even lines have been scanned, the uneven lines (dark) are scanned from the left and right camera alternately.

The impact of this modification is shown in Fig. 2.6. A small test was performed where the stereo camera setup was positioned at a fixed height above a large chessboard (to assure texture). A record was made of the camera stream while during the first few seconds the scene was static. After a few seconds, the chessboard was slid back and forth (left-right in the camera view) to introduce motion. The disparity was then computed for both types of camera implementations to see the effect of motion on the output. From the figure it is clear that the 'initial' system (top plot) performs significantly worse as soon as the scene starts to move. From the data one can see the left-right motion of the chessboard. When the chessboard slides to the left, the images appear to move towards each other. Hence a smaller disparity is measured. When sliding in the other direction, larger disparities are measured. From the top plot it can be seen that this motion is not visible from

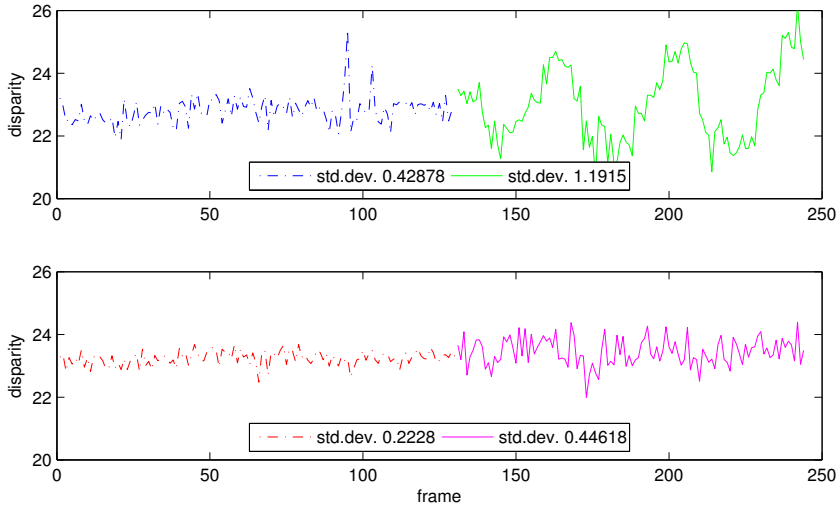

Figure 2.6: Comparison between the camera reading methods. **Top** initial method **Bottom** implemented method. During the first 135 frames there is no motion (dash-dotted lines), further on there is a relative motion between the camera and the chessboard (solid lines).

the measurements. But it should be noted that the measurements show smaller deviations during the first seconds of the experiment when there was no motion.

In this setup the effective resolution is reduced to a quarter of the original resolution. However, this is not an issue since the images are sub-sampled to a resolution of 160x108 to perform stereo processing at 25Hz. As noted before the camera images can be subject to noise. Furthermore, in the current setup both cameras make use of the same intensity calibration parameters, which only apply to one of them. As a result there is a major difference in sensitivity to bright image features. The cameras are also very sensitive to direct and reflected sunlight. This can blind the cameras. Also high frequency light sources can have a disturbing effect.

2.3. Analysis of existing stereo vision algorithms

Computer stereo vision is the extraction of 3D information from digital images. In general this implies that images from two or more cameras are evaluated by an algorithm that tries to compute which pixels correspond to the same physical object. When this matching is done, it is known for each pixel how large it is shifted in other images. By knowing the characteristics of the cameras, these shifts (denoted as 'disparities'), can be converted to real xyz-coordinates. By using all image pixels together a 3D reconstruction of the scene can be obtained.

A considerable amount of research has been done for decades on the problem of computational stereo vision. This research is still ongoing with focuses on quality

and computational efficiency. These are conflicting aspects. A concise overview of computer stereo vision methods that have been developed over the years is beyond the scope of this study. Interested readers are referred to the Middlebury taxonomy of Scharstein and Szeliski [13] and the evaluation of Tombari et al. [14] for overview articles. For stereo vision on a flapping wing MAV, the main requirement is implementability in real time systems. Real-time performance can be obtained in two ways: by using efficient algorithms or by using special hardware implementations. In this study, the focus lies on efficient algorithms. Using for example a Graphical Processing Unit(GPU), Field Programmable Gate Array (FPGA), Application-Specific Integrated Circuit (ASIC) or Digital Signal Processor (DSP) allows the use of optimized computation strategies that are very specific and have a limited applicability. Since the aim of this study is to converge to full autonomy, onboard processing is also a topic of interest. It is believed that if algorithms cannot be implemented on a CPU in real-time, they will also be no candidate for onboard processing in future systems. The focus in this study is therefore further limited to methods that enable real-time performance on CPUs.

Comparison Stereo vision algorithms can be divided in four groups depending on the optimization strategy they are based on: Winner-Takes-All, One-Dimensional Optimization, Multi-Dimensional Optimization, and Global Optimization. Fig. 2.7 shows a comparison among these types of optimization. Global Optimization is left out of this comparison because of its computational complexity. From each of the other three types, an example from the OpenCV library was taken to demonstrate the most important differences. The figure shows the result of each type of optimization method for the same image. The stereo images were sub-sampled such that each method had real-time performance. The parameters were tuned to obtain the best result.

The Block Matching method shows a relatively sparse result. Dominant features, such as vertical lines, are matched quite well, but in between these features a lot of unknown regions are left empty (black pixels). Even the shadows on the ground apparently do not provide enough texture for good matching. The information from this method is partly useful, in that it provides information on obstacles close by. But this information would be much more useful if the method would be able to indicate that the center zone of the image contains only obstacles far away. Note that the center zone even contains blobs of white pixels that indicate non-existing close objects.

The Dynamic Programming method performs even worse. The main structures in the image can not even be distinguished. This result might not be fully representative for dynamic programming algorithms since these perform better than winner-takes-all methods in general. In the top-left corner of the image the streaking effect is visible: the image lines appear as if they are a little bit randomly shifted horizontally (typical effect of Dynamic Programming). The bottom-left part of the image is almost empty (no reliable matches) and the right part of the image does not show clear objects. This illustrates the short-coming of Dynamic Programming: matching errors influence the results for the remainder of the image lines. The bad

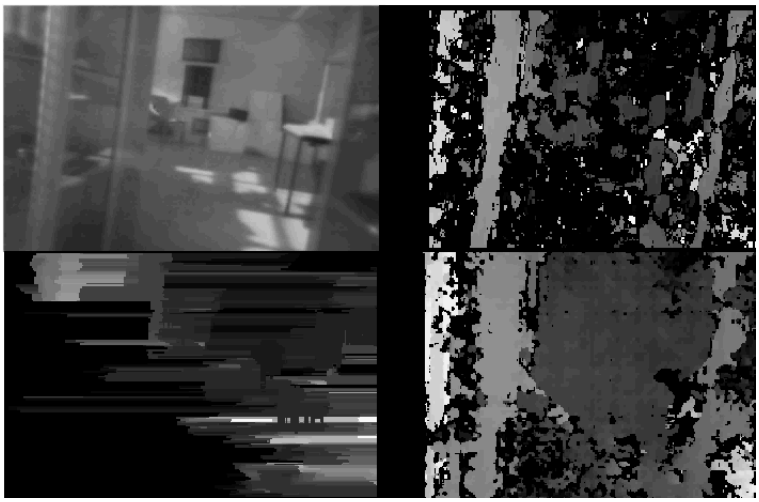

Figure 2.7: Comparison of three different types of stereo vision methods. **Top-Left** test image **Top-Right** Block Matching (Winner-Takes-All) **Bottom-Left** Dynamic Programming (One-Dimensional Optimization) **Bottom-Right** Semi-Global Block Matching (Multi-Dimensional Optimization)

matching results in the left part of the image spoil the results in the right part of the image. The fact that this implementation uses pixel-to-pixel matching costs might have a negative influence of the final result.

Compared to the other two methods, Semi-Global Block Matching gives significantly better results. The main structure of the scene is clearly visible in the disparity map: two cabinets close by on both sides and in between there is space with obstacles much further away. Also here some regions are left empty but the amount of known disparities is substantially larger. False matches are also visible but their number is also small. This method gives the most useful information, and is potentially useful enough for obstacle detection. The result is also notable because the method relies on simple pixel-to-pixel matching costs.

According to literature, Semi-Global Matching represents a good trade-off between computational efficiency and performance [15] [16] [17]. Based on the findings from literature and the above results that support these findings, it was decided to use the Semi-Global Matching [18] method for implementation in the obstacle avoidance strategies that were developed and tested in this study.

2.4. Performance analysis of the vision system

The performance characteristics in terms of distance measurements accuracy are discussed in this section. These are based on static and flight tests.

2.4.1. Static accuracy measurements

An important performance measure for the stereo vision system is its accuracy of measuring distances to objects. To measure its actual performance without the

influence of platform vibrations, a static test was done. For this test, the camera was fixed at several distances (100,150,200,250,300,400,500 and 600 cm) from a screen. The screen was a chess mat that was hanging vertically in the field of view of the camera. The stereo vision system was used in the same way as it is during flight. Disparity maps were computed from 1100 frames per measurement point. From each disparity map a small patch of 10x10 pixels was taken from the center of the map to compute the mean disparity. This disparity was used for calculating the distance from the camera to the screen. The results are shown in Fig. 2.8. From the results it can be observed that, at least for the static case, the stereo camera system is capable of measuring the distance to obstacles up to 5 m with a mean error of less than 50 cm. For the task of obstacle avoidance this can be regarded as an acceptable performance. Obstacles that are even farther away will be detected with a lower distance accuracy. The mean error is larger than 140cm in these cases.

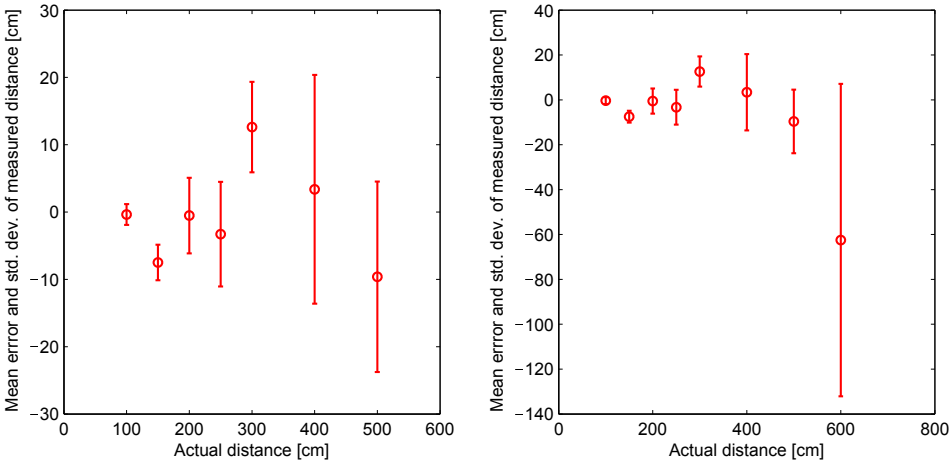

Figure 2.8: Distance measurement accuracy for the static case. The left plot is a detailed version of the right plot.

2.4.2. Accuracy measurements during flight

The accuracy of the stereo vision system has also been measured in flight. The experiment was performed using a free-flying DelFly II at a speed of approximately 60cm/s. The DelFly was flying in the direction of the chess mat. Two external cameras were used to track the position of the DelFly. Tracking was performed as follows: two video cameras were positioned such that the chess mat would be in their field of view and also the area in front of the chess mat (around 5m). The cameras were positioned on both sides of the flight path of the DelFly. By using a powerful background subtraction routine [19] and blob tracking, a special small marker positioned under the DelFly could be tracked. By using triangulation routines from OpenCV, the three-dimensional flight path (w.r.t the chess mat) of the DelFly

was determined. The measurements from the onboard camera and the external cameras were synchronized by looking for specific features in the recorded videos.

Fig. 2.9 shows the result from the first flight test. The blue points in the left plot indicate the distance between the DelFly and the mat, based on measurements from the external cameras. At small distances the blue points show some discontinuities. This is a result from the background subtraction. At small distances the DelFly flies between the cameras and the mat. The white marker on the DelFly will at some points not be noticed when it is in front of a white chessboard field. The tracking routine will then find another point on the DelFly, leading to triangulation errors. These measurement errors should therefore be ignored.

The red and green dots are onboard distance measurements. As can be seen from the plot, most of these points are concentrated around the blue points. However, some very clear outliers (red dots) are visible. These measurements result from a hardware problem. As a result some video frames received by the ground station are mixed-up. The order of the scan-lines is then different from the normal case and the left-right images going to the stereo processing routine contain wrong combinations: two images from the same camera, or swapped left-right images. This results in corrupt disparity maps. Another problem is a typical haze effect which results in images that are a mixture of two images.

These bad results (red dots) were left out by detecting and omitting corrupt frames. The curve fit is based on the good measurements (green dots). The right plot in the figure shows the deviation of the measurement points based on the curve fit. A running average (green dashed line) was computed based on the average error with a windows size of 21. Also the standard deviation for the static case is shown in the figure for comparison.

From the top plot, it can be observed that the tracked distances and the measured distances show a very good correspondence. The main observation from the bottom plot is that the onboard measurements have a larger standard deviation than those obtained during the static test. For distances larger than 350 cm, the error seems to grow rapidly, but this is at a moment that the DelFly is still turning towards the mat.

2.5. Flight test results using different obstacle avoidance strategies

This section discusses the results from tests with two different obstacle avoidance strategies.

2.5.1. Reactive yaw control

The turn logic for this strategy is straightforward. From the disparity map obtained by the stereo vision algorithm it is computed how many pixels belong to obstacles that are on short range (less than 1.1m). These pixels are summed separately for the left and right halves of the image, forming so-called 'obstacle-signals'. If the left obstacle-signal reaches a threshold a turn to the right is initiated, and vice versa. If both obstacle-signals reach the threshold at the same time, a right turn is initiated.

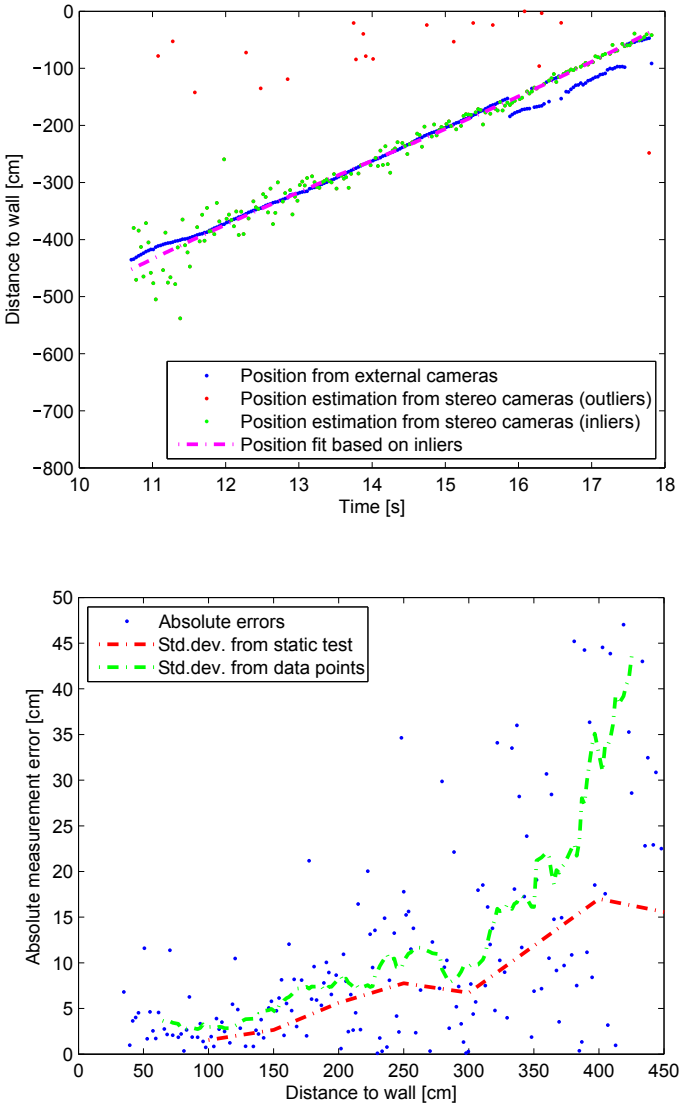

Figure 2.9: Distance measurement accuracy for the flight test. The top plot shows the actual distance and estimated distance over time. The bottom plot shows the estimation error with reference to the actual distance.

The threshold value has been chosen such that image noise and computational errors do not induce unnecessary turns. The turn is initiated by giving a predefined step input to the rudder. This rudder input is a fixed value that can be set separately for left and right turns. Its value was chosen such that the turns are steady and

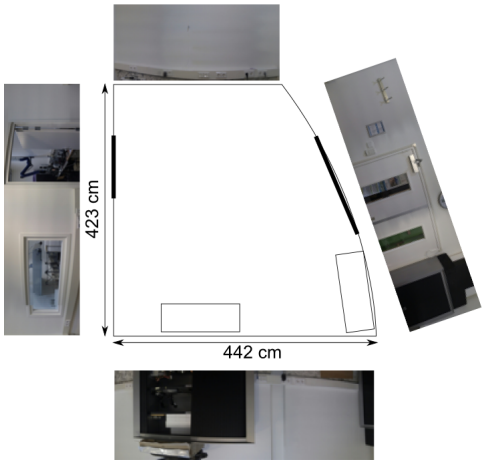

Figure 2.10: Floor plan of the test room. The images around show the walls, doors and cabinets in the room

symmetrical (around 40 cm radius) and the turn speed is not too fast to avoid spiral motions. The turn will end only as soon as both obstacle-signals become lower than another (and smaller) threshold. As soon as the lower threshold has been reached, the rudder will go back to its trim position. However, if one of the obstacle-signals reaches the higher threshold again within a predefined safety-time, the DelFly will continue its previous turn, regardless of which of the two 'obstacle'-signals reached the threshold. This will prevent the DelFly to turn back into the direction it just turned away from, since this is most likely not a safe maneuver.

The experiment was conducted in a room of ~4.23x4.42m. Fig. 2.10 shows a floor plan of the room. The images on the sides give a good impression of its appearance. Except for the walls, the main obstacles are two black cabinets. The door on the left was closed during the experiments, and part of the window on the left was covered to prevent window collisions. It should be noted that the images in the figure only show a part of the scene (mainly the top part) while the onboard cameras of the DelFly could see more of the lower parts of the room. The lights were most of the time switched off during the experiments since they resulted in a flickering effect in the stereo cameras. During the experiments the 'obstacle'-signals were logged, as well as turn events. Furthermore, an onboard image was captured at the moment a turn event (left/right turn or end of turn) occurred. The elevator was given a constant input such that the speed would be around 0.6m/s during the test.

This experiment was repeated several times and resulted in various observations. As a general result, it can be stated that the obstacle detection performed well. The obstacle avoidance strategy showed some expected flaws. This will be illustrated by data recorded during one of the flights.

Fig. 2.11 shows a situation during the first seconds of one of the test flights. The

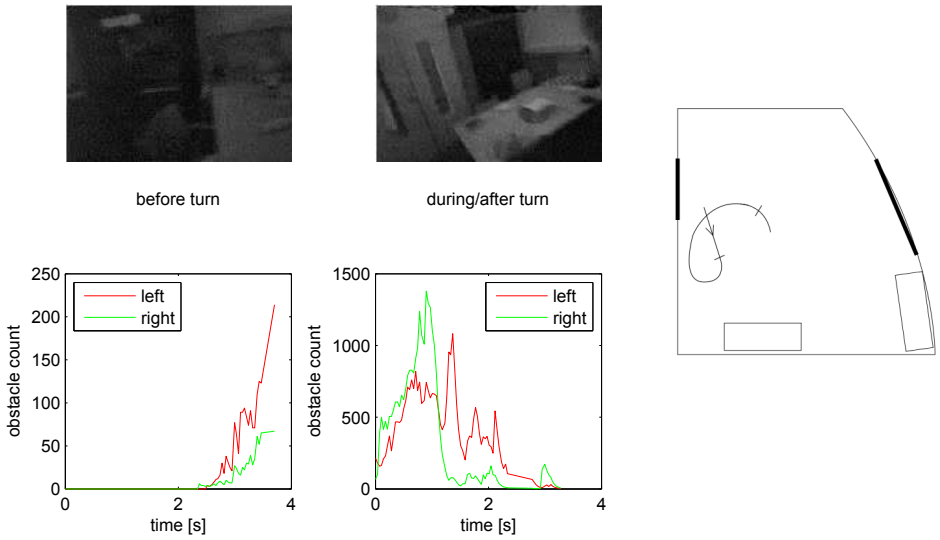

Figure 2.11: Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn.

sketch on the right indicates the position of the DelFly at the start of the flight and during the first turn. During the first seconds, it flies close to the wall. But, as can be seen in the left onboard image, the wall on the right is outside the field of view. The left bottom plot shows the 'obstacle'-signals during the last seconds before the turn. In this case these values are initially zero because the obstacle detection was not activated yet. From the plot it can be seen that the cabinet, (mainly) on the left side in the image, lets the left obstacle signal increase faster than the right signal, as expected. When the left signal exceeds the threshold (in this experiment set at 200), a turn to the right is initiated successfully. As a result, the DelFly turned into the direction of the wall. It was prevented from colliding manually.

The middle bottom plot shows the amount of obstacles detected during the turn. It can be seen that during the first half second of the turn, the amount of right 'obstacles' increases first. This is because the wall now enters the field of view. After one second the DelFly has turned around and the right obstacle signal decreases. Since the wall is now in the left side of the view, the left obstacle signal is now very high. While turning away from the wall, the left obstacle signal decreases. It can be observed that it takes fairly long before a safe flight direction was found during the turn. First it takes two seconds before the left obstacle signal decreases below the threshold (set at 50). If this threshold would have been the same (also 200) the turn would have been ended approximately one second earlier. However,

earlier experiments showed that for a threshold value of 200, turns would very frequently be ended too early (and then continued immediately, but with some delay). Also note that at the end of the turn, the left obstacle signal decreases below 50, but at the same time the right signal increases again. From the right onboard image in Fig. 2.11 it can be observed that the DelFly rolls while making a turn. The table in the image appears to be shifted up in the right side of the image. Apparently it is then detected as an obstacle.

In the sketch on the right it is indicated at which points the turn was initiated and ended. The end point corresponds to the location where the right onboard image (see figure) was taken. In the sketch it is indicated that after this moment the DelFly continued its turn a bit longer. This is a result from the delay between the ground station and the DelFly.

Fig. 2.12 shows how the DelFly continued after the first turn. It is flying into the direction of the same cabinet as before, but now it is in the right side of the camera field of view (top left image). The left bottom plot indicates that indeed an obstacle is detected on the right side. During the first 0.2s after turn initiation (middle bottom plot), the 'obstacle'-signals increase quickly since the DelFly approaches the cabinet. Then, after some delay, the turn command is received on board and the DelFly starts to turn to the left. Note that around one second later, both signals drop quickly. However, it takes another second before the signals drop below the threshold value. Apparently this is caused by the other cabinet in the corner. It should be noted that during this turn significant noise occurred. As a result, no obstacle detection was performed between 1.23s and 1.66s after turn initiation. This is also the case in the left bottom plot. In that case there are no measurements between 0.77s and 1.0s after the previous turn.

These examples show that the DelFly successfully detects obstacles in its field of view at sufficient range to perform obstacle avoidance. Also during the turns the obstacle detection provides reliable information which makes it possible to decide at which point the turn can be ended safely.

Situations as described in the first example can occur because of the direct nature of the turn strategy in combination with the limited field-of-view of the stereo cameras. During some of the experiments these situations occurred rarely and the DelFly could fly autonomously for longer than 1 minute.

An important observation during the tests is the endurance of the DelFly in its current configuration. As discussed earlier, almost full throttle needs to be applied right from the start of the flight. Within one minute, full throttle is required. Within 2-3 minutes the batteries cannot deliver sufficient power to keep the DelFly at a constant height anymore.

2.5.2. Reactive yaw and pitch control

The second experiment is a follow-up of the first one, and it was done the same way. The only difference is the addition of a simple pitch control rule. During unobstructed flight, the elevator is in its fixed position such that the DelFly will fly at a speed of around 0.6m/s. As soon as an obstacle needs to be avoided, a turn is initiated the same way as in the first experiment. At the same time, the elevator

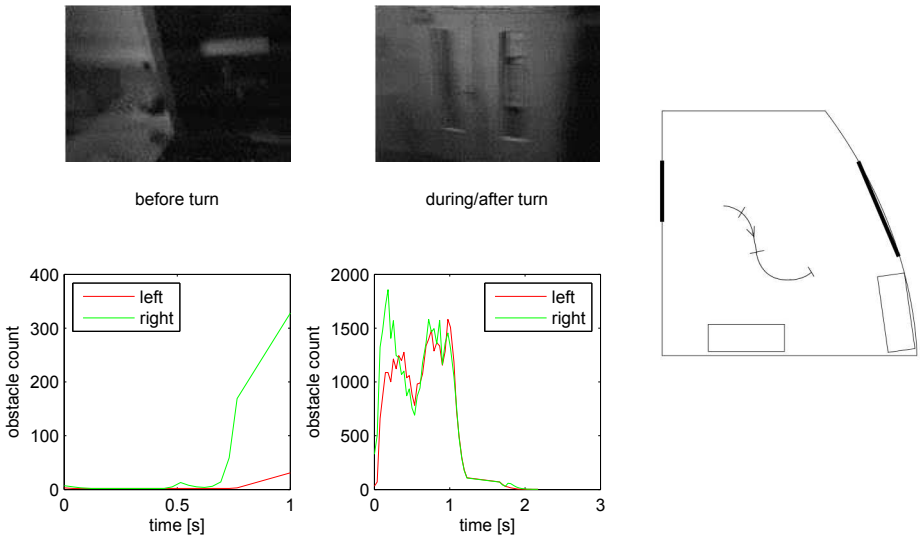

Figure 2.12: Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn.

input is changed such that the DelFly will loose its speed and start to hover. As a result, the DelFly will change its heading (by using the rudder) while it keeps its position. Obstacles can be avoided without the risk of making a turn and colliding with another object out of the camera field of view.

Before this test was conducted, it was already known that the DelFly in its current configuration is too heavy for hovering. It will definitely loose height at the turning points. However, the experiment can be useful in demonstrating that this simple avoidance strategy is suitable for an (FW)MAV as long as it is able to hover. Future designs of the DelFly might be able to hover more efficiently and could use this strategy for maneuvering in small spaces.

As explained, this second avoidance strategy is meant to demonstrate the benefit of making turns without forward speed. An example situation is shown in Fig. 2.13. The DelFly approaches the cabinet and at some point a turn is initiated. From the bottom middle plot it can be seen that initially the amount of left detected obstacles increases because of control delay and initial forward speed. The DelFly turns to the right and the obstacle-signals decrease. From the right onboard image (top middle image), it was observed that the DelFly has lost some height and is now flying slightly above table height. As discussed earlier, this is an expected (but unwanted) result due to the bad hover performance of this specific

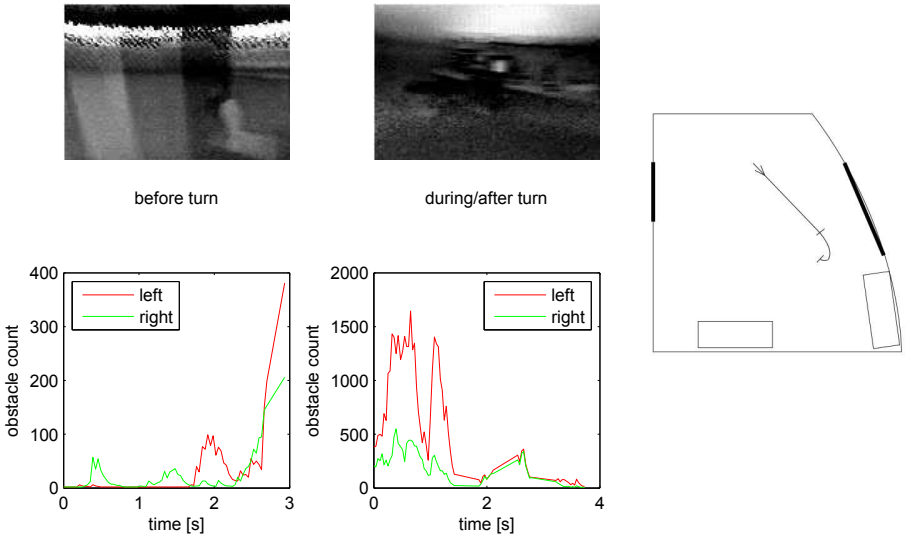

Figure 2.13: Example of a turn decision. The first image (top left) is an onboard image from the moment the turn decision threshold (200) was reached. The corresponding obstacle-signals (up to turn decision) are shown in the bottom left image. The other onboard image (top middle) was taken at the moment the lower threshold (50) was reached. The corresponding obstacle-signals (from turn initiation until end of turn are shown in the bottom middle image. The figure on the right shows the flight path during the turn.

configuration of the DelFly. Furthermore, this results in the additional problem that there are now other obstacles detected, such as those that are on the tables. Also partly because of noise (and bad obstacle detection measurements for that reason) the turn is ended at a point much farther than one would expect.

It was observed that with this strategy obstacle detection and avoidance could be performed successfully without the problem of colliding with out-of-sight obstacles. This demonstrates the advantage of stereo vision over optical flow measurements.

2.5.3. Short-term planning of yaw control

The previous experiments demonstrated that the DelFly is not able to hover. Furthermore, it was demonstrated that responding directly to obstacles in the field-of-view will result in collisions with obstacles that are outside the field-of-view. Therefore a third turn strategy is discussed here.

In this new strategy, the DelFly continuously flies with a constant speed (fixed elevator setting). A turn is initiated when too many obstacles (pixels with a large disparity value) are detected in a pre-defined safety region. The safety region is defined such that it covers an area large enough for the DelFly to turn around 360 degrees. Because of the limited field-of-view of the camera, this turn area lies ahead of the current position of the DelFly. Fig. 2.14 shows this safety region. The

region is defined in the camera reference frame, with the x-direction positive to the right, the y-direction positive up and the z-direction positive in the direction of flight. Starting at the origin (position of the camera), two oblique lines define the camera field-of-view. The dashed line is the trajectory the DelFly will follow as soon as too many obstacles are detected. After 225 cm, a right turn will be initiated. By making right turns only, only one safety region needs te be defined. During the turn, the same safety region is used to detect a new safe flight direction. As soon as it is found, the turn will be terminated. Because a turn might be terminated by mistake or an overshoot can occur due to delays, the turn will be continued immediately if the new direction of flight is not regarded safe anymore. This is only possible immediately after the previous turn was terminated (within one second). This type of turn recovery has been taken into account in the safety region definition. This is why the outer circle has been drawn. Around this outer circle extra safety margin has been included to accommodate for the width of the DelFly and inaccuracies in range estimations. Note that the turn area has been centered with respect to the field-of-view in order to minimize the total size of the safety region. As a result, the flight trajectory towards the turn area is drawn as a slanted line. For this reason, the stereo vision cameras need to be mounted on the DelFly with an offset angle (to the right) to align the drawn flight trajectory with the flight direction of the DelFly.

Note that this description only explains the conceptual idea behind this strategy. A comprehensive explanation and a further developed version of this strategy can be found in Sec. 4.4. This section also gives a detailed description of how the method is implemented.

In this strategy, only rudder commands are used. Because the obstacle measurements are sensitive to noise, filtering is required to increase robustness. For this reason a logical diagram was developed as shown in Fig. 2.15 which decides upon rudder inputs. In this logic each turn is divided in phases. During Phase 1, the DelFly flies straight with 1m/s (faster than during earlier experiments to increase flight endurance). A threshold is used for the turn decision based on the amount of pixels that exceed the disparity constraint defined by the safety region. To suppress noisy measurements, filtering is applied as follows. Each time the threshold is exceeded (250 pixels), the current time is stored. It can then be checked if the threshold is exceeded ten times within one second. If this is the case, it is concluded that there is an obstacle. The earliest detection time of these ten detections is then used as reference time for the second phase. In Phase 2, the DelFly still flies straight and waits until it has reached the point in Fig. 2.14 where the turn needs to be started. This time-to-turn is just over 2 seconds. However, because the turn response of the DelFly to rudder inputs is sluggish initially, and because of communication delays, the time-to-turn was tuned experimentally and set a a value of 1500ms. After this time has elapsed the turn is initiated in Phase 3. During the turn it is checked if the current direction of flight is obstacle free. As soon as a lower threshold of 200 pixels is reached, Phase 4 starts. No filtering is used here because this will result in unwanted delay. The turn speed of the DelFly is around 1 rad/s and small delays result in large flight direction differences. To compensate for the quick decision making and turn overshoots, it is checked in Phase 4 if the

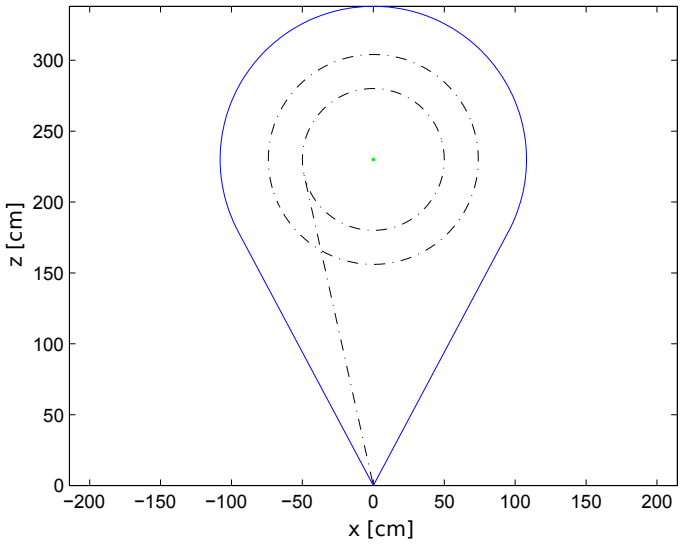

Figure 2.14: Turn strategy using continuous turns. The dashed line is the DelFly flight trajectory. The area between the blue line is the obstacle-free region.

new flight direction is indeed a safe direction to fly. If within one second after turn termination the obstacle threshold of 250 pixels is exceeded again, the turn is resumed in Phase 3. Otherwise, the new flight direction is regarded as safe and Phase 1 starts again. In Phase 1, also another check is performed to detect obstacles at short range. The avoidance region defined as in the first experiment is used here. A threshold of 500 pixels is used. If it is exceeded three times in a row, Phase 3 is activated to start turning immediately. The main reason for including this rule is the sensitivity of the DelFly to wind disturbances which changes the flight trajectory to such an extent that obstacles initially out of the field-of-view will result in collisions. This rule is used to prevent unexpected collisions but does not guarantee that the DelFly will be able to continue safely. Tests with this turn strategy were performed in a larger test room because of the size of the safety region. In the test room from the first experiment, the DelFly would keep turning continuously. The test room is visualized in Fig. 2.16.

Fig. 2.17 shows the result from the test with the best result that has been obtained. During this test, the DelFly flew around for 72.6s without hitting any object. The experimenter did not have to pull the DelFly up to keep it at a constant height. It should be noted that the experiment was ended without reason. The DelFly was still performing autonomous flight and the total successful test time could have been longer than the reported length.

The experiment starts at a point where the DelFly is coming out of a turn that was performed early after start up. This is point t=0 in Fig. 2.17. The track colors indicate the flight phases. The controller uses the logic according to Fig. 2.15.

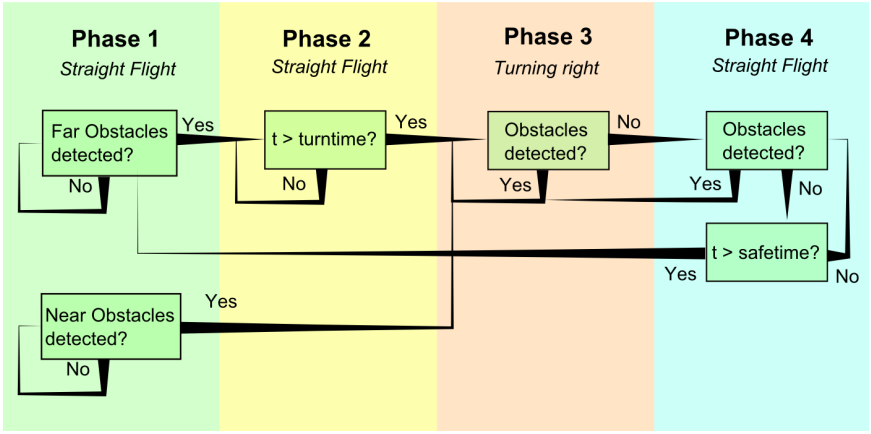

Figure 2.15: Flight phase diagram showing the different flight phases, the decision making process and the rudder command for each phase.

During start of the experiment, Phase 4 is active where the DelFly is ending its turn. During the next ten seconds, Phase 1 is active and the DelFly should perform straight flight. From the flight track it can be observed that the flight is far from straight. Due to non-zero wind speeds in the test room, caused by ventilation and air conditioning systems, the DelFly swerves significantly. At this time, this does not result in avoidance problems. When the DelFly approaches the upper wall after 10 seconds, Phase 2 and 3 are activated. These phases share the same color in the figure. The flight track during the subsequent Phase 4 goes right past the cabinet. However, at this point the DelFly was flying above cabinet height, and for that reason it was not considered to be an obstacle. Only the wall needed to be avoided at this point. The flight is then continued in the direction of the lower wall. Note that in all cases where Phase 4 is active, the turn continues. In many cases the new flight direction is around 90 degrees further to the right. This is a result from all system delays, including video reception delay, processing time and radio control delay. After 20 seconds in the experiment Phase 1 is active again. Note that the flight track unexpectedly deflects to the left. As a result a new turn is triggered which directs the DelFly back to the upper wall. This sudden left turn can be explained by yaw/roll instability and is unpredictable. The cabinets in the top then force the DelFly to go back (at around 30 seconds). Note that back at the bottom wall the DelFly preserves a larger distance to the wall compared to other turns. Again the DelFly goes to the top cabinets and back. Just after 50s the lower wall is approached again. In this case an early turn is initiated which is ended too early. As a result Phase 4 is activated while the DelFly still continuous in the direction of the wall. Because the wall is detected again Phase 3 is active again after 684ms. The turn is then continued till the flight direction is now in the direction of the cabinets again. Again the DelFly unexpectedly turns quickly to the left and flies in the direction of the doors on the left. These are detected early and a

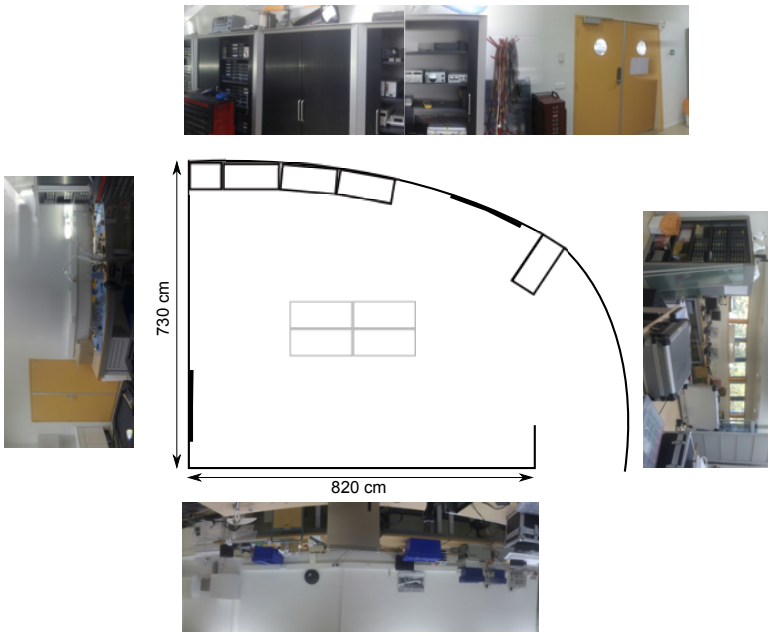

Figure 2.16: Floor plan of the test room. The images indicate the type of texture present along the walls of the room.

slight turn follows immediately. Another turn is then initiated 1517ms later to avoid the left wall. At t=60, the DelFly is performing a straight flight in the direction of the top wall at a height above the cabinets. The wall behind the cabinets is then detected and avoided successfully. The flight ends after 72.6s without colliding with any obstacle. Note that during the last part of the flight the DelFly gradually but severely makes a turn to the right. Near obstacles (see bottom of Phase 1 in Fig. 2.15 were never detected. This means that the DelFly never tried to avoid obstacles that were detected late.

2.6. Conclusions

From the results presented in this study it can be concluded that stereo vision can be applied successfully for obstacle detection and avoidance on FWMAVs. It was shown that real-time stereo vision can provide accurate and sufficient obstacle information. By making use of suitable camera hardware, the flapping motion of FWMAVs has a minor influence on the stereo vision algorithm. In this respect, this method outperforms optical flow techniques.

The small camera system is capable of giving distance estimates with a standard deviation of 20 cm at a range of 5 m. Even for texture-poor areas, the accuracy is still adequate. The weight of the camera system and extra required battery leads

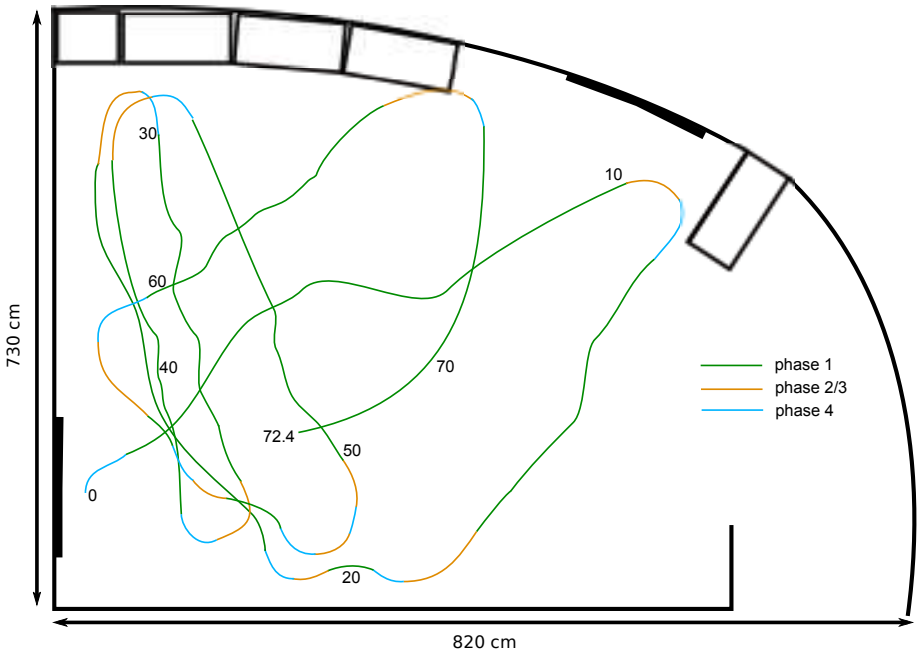

Figure 2.17: Flight track of the DelFly during the experiment. The numbers indicate the flight time in seconds, the colors represent the flight phases.

to a reduced flight endurance and a reduced flight envelope i.e. hovering is not possible.

Closed-loop experiments showed that stereo vision can provide robust and reliable obstacle information that allows the DelFly to perform successful obstacle avoidance. An autonomous flight time of 72.6 seconds with no collisions has been obtained as the best result.

One of the focuses of future research will be on the camera design. Lighter cameras with a wider field of view should result in better performance. Another important focus will be on onboard image processing. This will eliminate communication delays and the need for a ground station within communication range.

References

[1] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, *Obstacle and terrain avoidance for miniature aerial vehicles,* in *Advances in Unmanned Aerial Vehicles* (Springer, 2007) pp. 213–244.

[2] C. Kownacki, *Guidance and obstacle avoidance of mav in uncertain urban environment,* in *European Micro Aerial Vehicle Conference and Flight Competition* (2009).

[3] L. L. Hines, V. Arabagi, and M. Sitti, *Free flight simulations and pitch and*

roll control experiments of a sub-gram flapping-flight micro aerial vehicle, in *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (IEEE, 2011) pp. 1–7.

[4] S. H. Lin, F. Y. Hsiao, C. L. Chen, and J. F. Shen, *Altitude control of flapping-wing MAV using vision-based navigation,* in *American Control Conference (ACC), 2010* (IEEE, 2010) pp. 21–26.

[5] P. E. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, *Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor.* in *ICRA* (2012) pp. 4228–4235.

[6] S. S. Baek and R. S. Fearing, *Flight forces and altitude regulation of 12 gram I-Bird,* in *IEEE RAS and EMBS Int Conf on Biomedical Robotics and Biomechatronics (BioRob)* (2010) pp. 454–460.

[7] S. S. Baek, F. G. Bermudez, and R. Fearing, *Flight control for target seeking by 13 gram ornithopter,* in *IEEE/RSJ Int Conf on Intelligent Robots and Systems.* (2011).

[8] F. G. Bermudez and R. Fearing, *Optical flow on a flapping wing robot,* in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2009) pp. 5027–5032.

[9] R. Tedrake, Z. Jackowski, R. Cory, J. W. Roberts, and W. Hoburg, *Learning to fly like a bird,* in *14th International Symposium on Robotics Research. Lucerne, Switzerland* (Citeseer, 2009).

[10] G. C. H. E. de Croon, K. M. E. de Clerq, R. Ruijsink, B. D. W. Remes, and C. de Wagter, *Design, aerodynamics, and vision-based control of the DelFly,* International Journal on Micro Air Vehicles **1**, 71 (2009).

[11] 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 **28**, 529 (2012).

[12] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[13] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,* International Journal of Computer Vision **47**, 7 (2002).

[14] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, *Classification and evaluation of cost aggregation methods for stereo correspondence,* in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (IEEE, 2008) pp. 1–8.

[15] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, *Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation,* in *Embedded Computer Systems (SAMOS), 2010 International Conference on* (IEEE, 2010) pp. 93–101.

[16] S. K. Gehrig and C. Rabe, *Real-time semi-global matching on the cpu,* in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (IEEE, 2010) pp. 85–92.

[17] B. Chen and H.-p. Chen, *A realization of semi-global matching stereo algorithm on gpu for real-time application,* in *Seventh International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR2011)* (International Society for Optics and Photonics, 2011) pp. 80040W–80040W.

[18] H. Hirschmuller, *Stereo processing by semiglobal matching and mutual information,* IEEE Transactions on pattern analysis and machine intelligence **30**, 328 (2008).

[19] O. Barnich and M. Van Droogenbroeck, *Vibe: A universal background subtraction algorithm for video sequences,* IEEE Transactions on Image processing **20**, 1709 (2011).

3

The DelFly Explorer: a Flapping Wing MAV with Onboard Stereo Vision and Processing

The previous chapter demonstrated that stereo vision is able to outperform optical flow in doing obstacle avoidance. The presented flapping wing system relies on off-board processing of the stereo vision images and therefore cannot be considered to be fully autonomous.

In this chapter, fully autonomous obstacle avoidance is performed by the DelFly Explorer, a flapping wing MAV that features an onboard stereo vision system including processing. A detailed description of this system is presented. Since the processing capabilities of such a small embedded system are very limited, a major focus of this study is on the implementation of an efficient stereo vision algorithm. It is also discussed how sub-sampling can be used as a trade-off between performance and efficiency. Furthermore, different obstacle avoidance algorithms are implemented and tested in real flights.

This chapter is based on the following article:
C. De Wagter, **S. Tijmons**, B.D.W. Remes, G.C.H.E. de Croon, Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System, In: IEEE International Conference on Robotics and Automation (ICRA), (2014)

3.1. Introduction

Autonomous flight of Micro Air Vehicles (MAVs) is a major challenge in the field of robotics. The low weight and small size of MAVs limit the sensors and processing that can be placed on board, while flying in environments with obstacles requires quick reactions. Impressive results in this area have been obtained with quadrotor MAVs (weighing in the order of 750 g), using sensors ranging from miniaturized laser scanners [1, 2] to RGB-D devices [3, 4] and stereo vision [5, 6].

Flapping Wing Micro Air Vehicles (FWMAVs) form a specific group of MAVs with the advantages of high maneuvrability, quick transition between multiple flight regimes and robustness to impact. Existing FWMAV designs typically have a weight in the order of grams. For example, the extremely small 'Robobee' weighs 0.8 grams (without onboard energy source) [7], and the 'Nano hummingbird' weighs 19 grams [8]. Although most research on FWMAVs focuses on their aerodynamics and design (cf. [9]), several studies have addressed various forms of autonomous flight, with varying sensor / processing configurations.

In the brief overview of related work below, four levels of autonomous flight are discerned as studied for FWMAVs: attitude stabilization, visual servoing, height control, and obstacle avoidance. Since obstacle avoidance has not yet been solved, higher level navigation has not yet been studied.

Attitude stabilization is only relevant for tailless FWMAVs, since they are passively unstable. Active attitude stabilization of a tailless FWMAV was first achieved by the Nano hummingbird [8] with onboard sensing and processing. Recently, attitude and 3D-position control has also been achieved with the Robobee by utilizing an external motion tracking system [7]. Visual servoing tasks have been performed in various studies with either off-board sensing and computing [10–13], onboard sensing and off-board computation [10, 14–16], or with onboard sensing and computation [17]. In [17] the camera and chip from a 'Wii-mote' were used for detecting and flying toward an infrared light. Height control with external cameras has been achieved by multiple platforms [10–13]. Vision-based height control in known environments with an onboard camera and off-board processing has been achieved in [10, 14], while height control based on an onboard barometer and processing has been achieved in [18]. Obstacle avoidance has been addressed in [15, 16, 18]. In [15, 18] obstacle avoidance was performed with a single onboard camera, while a laptop determined optical flow and a complementary 'appearance variation cue'. The success of monocular obstacle avoidance remained limited, with a typical flight duration in normal office rooms of around 30 seconds. In Chapter 2 stereo vision has been studied for obstacle avoidance with FWMAVs, demonstrating autonomous indoor flight. The processing was performed off-board.

For autonomous flight in unknown environments, onboard processing and exteroceptive sensing are essential. Moreover, the exteroceptive sensing needs to provide sufficiently rich information to allow for obstacle avoidance, and later, navigation. None of the above-mentioned studies fulfills these requirements.

This study presents the first FWMAV that performs onboard vision processing for autonomous flight in unknown environments. The DelFly 'Explorer' is a 28 cm wing span, 20 gram FWMAV equipped with a 0.98 gram autopilot and a 4.0 gram

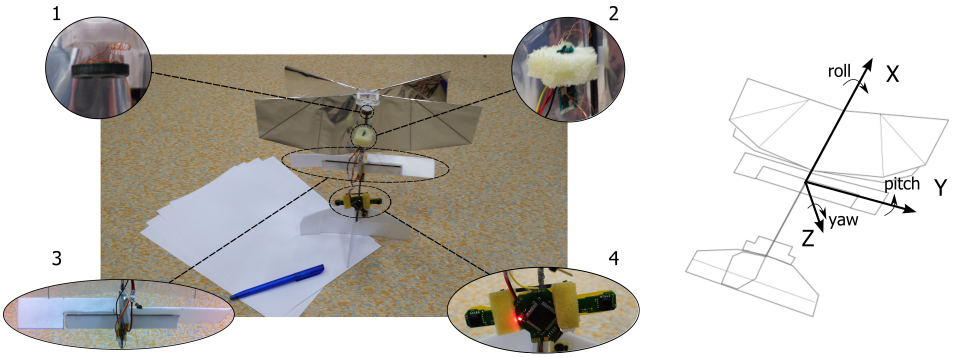

Figure 3.1: **Left:** Picture of the *DelFly Explorer*. The four insets show the main changes with respect to the DelFly II: (1) the number of windings in the brushless motors has been reduced to cope with the Explorer's higher weight, (2) an autopilot with a complete IMU, barometer, and an ATmega328P - MLF28 microcrontroller, (3) the DelFly explorer uses ailerons behind the wings instead of a rudder on the tail, and (4) the onboard stereo vision system with STM32F405 processor for onboard vision processing. **Right:** Sketch of DelFly Explorer in flight with the body-axes definition.

onboard stereo vision system. The main contributions of this study are: (1) the lightweight electronics for autopilot and stereo vision system, (2) the design improvements leading to more lift and better handling qualities for making turns with tailed FWMAVs (Section 3.2), and (3) a novel real-time and memory-efficient stereo vision algorithm, named LongSeq, which is robust to the FWMAV's flapping motion (Section 3.3). The onboard stereo vision system is illustrated in the context of an obstacle avoidance task in an environment with sparse obstacles (Section 3.4). Conclusions are drawn in Section 3.5.

3.2. System design overview of the DelFly 'Explorer'

In the course of the DelFly project, many DelFly versions have been created. The project started in 2005 with the *DelFly I*, which weighed 21.00 grams and had a 50 cm wing span. The *DelFly II*, was demonstrated in 2007. It had a 28 cm wing span, and weighed 16.07 grams. The smallest DelFly version, the *DelFly Micro*, was presented in 2008. It weighed 3.07 grams and had a 10 cm wing span. It is important to note that the unique property of the DelFly versions is not their size or weight, but that they can perform free-flight with onboard energy source and camera. The camera allows research on the use of FWMAVs as observation platforms or as autonomous robots.

Despite the weight of an onboard camera and transmitter, the DelFly II has a large flight envelope: it can fly forward at 7 m/s, it can perform slow forward flight, and it can even fly backward at -1 m/s. Although the DelFly II was presented to the public in 2007, its design and aerodynamics have been the subject of extensive study, leading to considerable improvements in the handling properties, possible lift, and flight duration. As a consequence of these properties and the miniaturization of electronics, more sensors have been added to it over time. While at first

the onboard images were processed off-board both for height control and obstacle avoidance [14, 15], in a more recent study height control was performed by an onboard microcontroller and barometer [18]. However, visual obstacle detection was still performed off-board, as described in Chapter 2.

The newest DelFly can carry sufficient payload to carry a 0.98-gram autopilot and a 4.0 gram stereo vision system (cameras and processor), the details of which are given below. Although the payload makes the DelFly heavier, it also allows the autonomous exploration of unknown spaces. Since this sets the newest DelFly apart from all previous versions, it is named as: the *DelFly Explorer*. It has a wing span of 28 cm and a weight of 20 grams.

The DelFly Explorer is shown in Fig. 3.1, with insets showing its four main innovative components. The first inset shows the brushless motor (Mighty Midget micro Nano motor, 10/1/36T). The number of windings around the coils has been reduced manually from 36 to 32. This way the ratio of rpm (and hence lift) versus input voltage is increased at the cost of a lower torque. As a result, the lift generated at 3.5V is still sufficient to keep the heavier DelFly Explorer in the air. This is in contrast to the old case where it would descend when the voltage dropped below 3.9V. The flight time of the DelFly Explorer is typically around 10 minutes.

The second inset shows a side-view of the autopilot, including an ATmega328P - MLF28 microcontroller, 3-axis accelerometers, gyros, magnetometers, and a barometer. Furthermore, it features two-way telemetry and rpm-monitoring. The autopilot is not necessary to achieve stable flight, as the tail of the DelFly passively stabilizes it during flight. However, the autopilot can serve other purposes, such as performing height control, disturbance rejection or more precise attitude control.

The third inset shows a set of ailerons placed just behind the wings. These ailerons are necessary for making smooth turns, which is essential to autonomous flight. The DelFly II featured a rudder for making turns. Deflection of the rudder first caused the DelFly II to yaw (around the Z-axis - see the right part of Fig. 3.1 for the axes definition), which in turn also resulted in a heading change. However, the yaw rotations during turns rendered computer vision processing during turns problematic. The ailerons of the DelFly Explorer make the DelFly roll (around the X-axis), and since it flies close to up-right, this directly influences the heading without creating any rotations of the camera images.

Finally, the fourth inset shows the stereo vision system in more detail. It has two digital cameras with a baseline of 6.0 centimeter and an STM32F405 processor. Importantly, the flapping motion of FWMAVs introduces deformations in the camera images [15, 19]. Therefore, it is not possible to use subsequently recorded left and right images for stereo matching, as mentioned in Sec. 2.2.2. The cameras of the stereo system are synchronized and provide $YUYV$ image streams, and in the current implementation a CPLD merges the streams from both cameras by alternately taking the Y component of the stream from both cameras. This results in a single image stream with the order $Y_l Y_r Y_l Y_r$. The resulting stream contains simultaneously sampled pixels at full camera resolution but without color.

3.3. Stereo Visions Algorithms

For the stereo vision system carried on board the DelFly, a new stereo vision algorithm was developed that is presented in this section. For autonomous obstacle avoidance of the DelFly, it is required to have real-time processing of the stereo images in combination with sufficient depth quality. Since the stereo system is heavily restricted in terms of processing speed (168 MHz) and memory availability (max. 192 kB RAM), it is important to find the right point on the trade-off between speed and quality.

Among the huge amount of stereo vision algorithms that can be found in literature, there are two groups that are not regarded to be suitable for the current application. These are the algorithms that perform global optimization, and the algorithms that are based on local matching. The first group is too demanding in terms of power and memory requirements, while the second group provides insufficient quality when dealing with image regions that contain little texture. In between these groups there is another group of algorithms that perform semi-global optimization. Examples of these algorithms are 1-D Dynamic Programming [20] and Semi-Global Matching [21]. These algorithms perform optimization along certain individual directions. The drawback of such an algorithm is that an error somewhere along this optimization line has an effect on the rest of the optimization line. These effects are limited in [21] by optimizing over multiple directions. However, this increases the required amount of processing and memory again.

3.3.1. LongSeq: an efficient stereo vision algorithm

For these reasons, a new algorithm is proposed that performs optimization along one image line at a time, where badly matched pixels do not have a degrading effect on the matching quality. For reasons to become clear in the explanation, the algorithm is called the *LongSeq* algorithm. The first step in the algorithm is to compute the matching costs $C(x,d)$ of the pixels in one image line by calculating the absolute difference in intensity between the left image I_l and the right image I_r for a disparity range d_{range} starting from a minimal disparity d_{min}.

$$C(x,d) = |I_l(x) - I_r(x-d)| \tag{3.1}$$

Then the minimum matching cost $C_{min}(x)$ for each pixel is computed:

$$C_{min}(x) = \min_d C(x,d) \tag{3.2}$$

Based on these cost measures (matching cost and minimum matching cost), a binary image B is computed for all pixels and disparities of the image line using two thresholds: τ_{cost} and τ_{min}:

$$B(x,d) = \begin{cases} 1 & \text{if } C(x,d) > \tau_{cost} \text{ and } C_{min}(x) < \tau_{min} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

The cost threshold τ_{cost} is used to define if a pixel match is good or bad. A matching cost above the threshold indicates a bad match. The minimum cost

threshold τ_{min} is used to check if there is at least one disparity value for which the pixel has a good match. $B(x,d)$ will only be nonzero when pixel x has a some good matching candidate, but if that is not the case for the disparity value considered. Pixels that have no good matching candidates are simply ignored. As a result, image B indicates which pixels have better candidates at other disparities. All other pixels are ignored at this stage since they have either no good matching candidate, or they match well at the considered disparity value.

The next step is to find sequences of neighboring pixels in an image line that do not have better matching candidates at other disparities (i.e. $B(x,d)=0$). The length of this sequence will be used as a measure for matching quality and it is therefore stored in image B. This is done by replacing all zero values by the length of the sequences they belong to. For example, let us consider eight neighboring pixels (50 to 57) in a line for one disparity value, e.g., 7. From Equation 3.3 the following fictitious values were obtained:

$$B([50\ 57],7) = [1\ 0\ 0\ 0\ 1\ 0\ 0\ 1]$$

This series of values contains two sequences of zeros; one with length 3 and one with length 2. The zeros in B then are accordingly replaced by these numbers.

$$B([50\ 57],7) = [1\ 3\ 3\ 3\ 1\ 2\ 2\ 1]$$

An initial disparity map D_{init}^{left} is then computed by selecting from B for all x the disparity value with the highest number (longest sequence):

$$D_{init}^{left}(x) = \max_{d} B(x,d) \tag{3.4}$$

The matching cost as described in Equation 3.1 is defined for matching the left image with the right image. The process is repeated for matching the right image with the left image to obtain D_{init}^{left} and D_{init}^{right}. These disparity maps can now be combined to optimize the result. This is done by mapping the left disparity image to the right disparity image:

$$D_{map}^{left->right}(x - D_{init}^{left}(x)) \leftarrow -D_{init}^{left}(x) \tag{3.5}$$

The optimal disparity is then found by taking the minimum of the two disparity maps:

$$D_{opt}(x) = \min(D_{map}^{left->right}(x), D_{init}^{right}) \tag{3.6}$$

This optimization step is required to handle disparity discontinuities. The algorithm is named LongSeq, because it favors long sequences with constant disparity in an image line. In situations where there is little to no texture, this will slightly bias the result to high-disparity estimates. In the context of obstacle avoidance, this is very sensible: low-texture images often occur close to obstacles and in any case present a danger, since they do not provide information on distances to obstacles ahead.

This method assumes that the images contain only fronto-parallel planes. Furthermore it specifically tries to match image planes with low variation in texture. By sliding these planes over each other, there will be one disparity where the overlap between the planes from the left and right image will reach its maximum. This effect is measured by the length of the sequences, and for this reason the maximum length is selected as the best match.

The proposed method shows some similarities with plane sweeping algorithms [22] in that it tries to match an image plane for a certain orientation. However, in the proposed method, only fronto-parallel planes are considered for computational reasons. Moreover, in contrast to [22], LongSeq searches the largest line section meeting this assumption.

3.3.2. Sub-sampling: increasing the computational efficiency

In the interest of computational efficiency, typical stereo vision steps such as undistortion and image rectification are skipped. Without these steps, LongSeq takes around 90 ms of processing on the STM32F405 on a full image of 128×96 pixels. Hence, it runs at ~ 11 Hz. For many applications of the stereo vision system 11 Hz can be sufficient. However, for some applications, such as obstacle avoidance or flying through a window, a higher processing frequency may be desired. The same goes if one wants to perform additional vision tasks besides stereo vision.

If the interest is not in dense 3D scene reconstruction, but some type of aggregate disparity values are used (as in Sec. 2.3), then *sub-sampling* can be applied. Sub-sampling typically leads to a considerable gain in computational efficiency at a low cost in accuracy [23]. Since LongSeq is line-based, a natural way of sub-sampling is to process fewer lines. The use of sub-sampling with the stereo vision algorithm will be tested in an application of sparse obstacle avoidance.

3.4. Application of the system to the obstacle avoidance task

The DelFly Explorer is applied to a sparse obstacle avoidance task. In the context of this task, results are shown of the stereo vision processing, also when combined with sub-sampling. Avoidance of sparse obstacles is rather straightforward, as is the employed control strategy. The main goal here is to show that the stereo vision system works in real-time and can cope with the FWMAV's flapping motion.

Specifically, the task of the FWMAV derives from the indoor competition of the IMAV 2013[1], which took place on September 19, 2013. The FWMAV had to take off autonomously and fly through a sparse obstacle field, keeping its heading. The obstacles are tall, brightly colored poles. This section first discusses the stereo vision results (Subsection 3.4.1), then explains the control algorithms involved in the experiment (Subsection 3.4.2), and finally presents the results of the experiment (Subsection 3.4.3).

[1] http://www.imav2013.org/

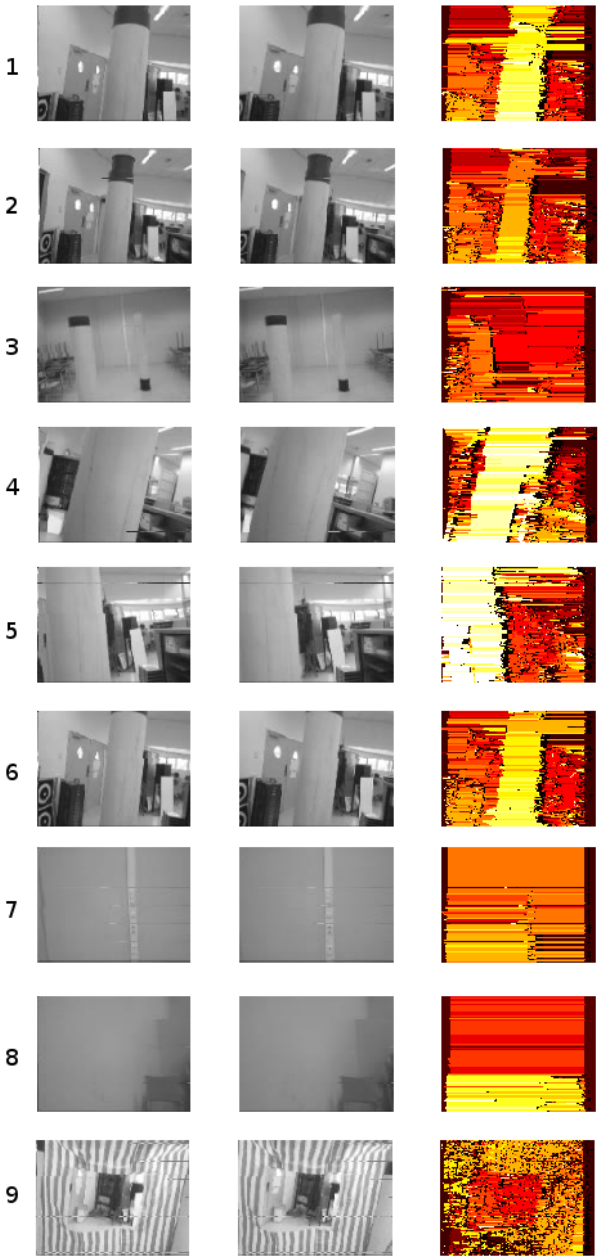

Figure 3.2: Nine examples of the stereo vision processing. The columns show from left to right: the left image, the right image, and the disparity image produced by the proposed stereo vision algorithm LongSeq. The disparity images are color coded from low-disparity (dark) to high-disparity (bright).

3.4.1. Analysis of the vision system performance

The stereo vision system on board the DelFly Explorer does not yet have any wireless connection for sending images during flight. Therefore, results are shown of the stereo system in-hand, with the images sent via a serial connection. Figure 3.2 shows nine examples of stereo vision images and their corresponding disparity maps. The left column shows the left images, the center column the right images, and the right column shows the disparity maps, in the interval $[0, 10]$ (bad pixels are also set to 0). Please remark that even though the camera is held in hand, the images already have motion deformations and blur. The top six examples show results for detecting poles as is the interest of the current application, while the bottom three examples show results in different situations.

The line-wise matching strategy of the proposed algorithm can be clearly seen in the images by the striping effects. By observing the detected poles it can be noted that texture poor areas tend to have the same disparity as the poles. This effect might be reduced in some cases by using more complex algorithms that perform optimization in more directions. For the task of avoiding poles this effect is not a real issue, since the pole will be avoided anyway. In general the background will be assigned the same disparity as the pole and not the other way around. Exceptions occur in situations where the contrast between the pole and the background is very low. The first six examples in Fig. 3.2 show that the presence of the poles is clearly indicated.

The effect of the pixel-based matching cost is illustrated by example 9 of Fig. 3.2. The dense variation in contrast in combination with the low resolution images results in many small sequences and a large variation of disparity values. This effect might be reduced by using windows for calculating the matching cost but this increases computational load as well as memory requirements.

The results from example 7 and 8 in Fig. 3.2 are far from perfect, but the results are useful for the current application. Example 7 shows that the wall is fairly close to the camera, even though the structure in the middle is the only feature that provides sufficient texture. In the case of example 8, the algorithm is able to indicate that the bottom part of the images contain obstacles at at smaller range compared to the rest of the image.

The control algorithm explained in the next subsection bases its decisions on the number of pixels with a disparity higher than 5 in the left and the right part of the image. This implies that the detailed disparity maps are aggregated into only two values. Hence, it makes sense to apply sub-sampling for achieving higher processing frequencies. Fig. 3.3 shows the number of processed image lines vs. the processing times as measured on the STM32F405 (crosses). As to be expected, this relation is roughly linear (dashed line). In order to process at frame rate, one can sample 32 image lines (one third of the image).

Fig. 3.4 shows the effects of sub-sampling on the estimated number of pixels with a disparity larger than 5 (top) and on the difference between the left and the right image (bottom). As can be seen in the figure, all sampling ratios follow the trend of the case of full sampling (100%) - albeit with a variation that increases with a decreasing sampling ratio. Surprisingly, this is even valid for a low sampling

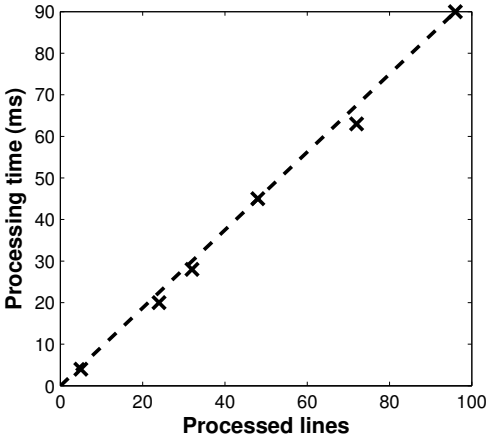

Figure 3.3: Processed image lines vs. processing time on board the DelFly.

ratio of 5% (4 image lines out of 96 in the current implementation).

3.4.2. Methods to use onboard sensors for flight control

This subsection discusses the control algorithms used for take-off, height control, and obstacle avoidance. Take-off is performed with open-loop control. Before the control sequence starts, the barometer measurement at that moment is taken as a reference for a height of 0 m. The sequence starts by setting the flap frequency above the trim setting, which results in a steep climb. After that the flap frequency is reduced to a trim value (for trimmed horizontal flight) after which closed loop control is performed on the height using the barometer feedback.

Obstacle avoidance is performed on the basis of the LongSeq's stereo vision processing. First, the number of pixels with disparity larger than 5 are determined in the left and right part of the image. If the total number of such pixels is lower than the empirically set threshold of 300, the DelFly will continue to fly straight. Else it will turn toward the side that has fewer such pixels, with a fixed aileron deflection. Sub-sampling is applied with 32 image lines, so that the processing matches the frame rate of the digital cameras.

3.4.3. Flight test experiments using reactive control

The DelFly Explorer can successfully take off and fly through an obstacle field. Fig. 3.5 shows the flight trajectories from a test with four trials (blue lines). In three out of four trials, the DelFly passes through the field without touching any obstacle. The trial with the dash-dotted line shows a track where the DelFly passed through the obstacle field, but touched two poles with its wings, resulting in sharp turns. In order to show that the vision actively determines the DelFly's heading, one trial is included where a gain is selected that inverts the avoidance reactions (red dashed trajectory). As a result, the DelFly does not avoid but targets the pole.

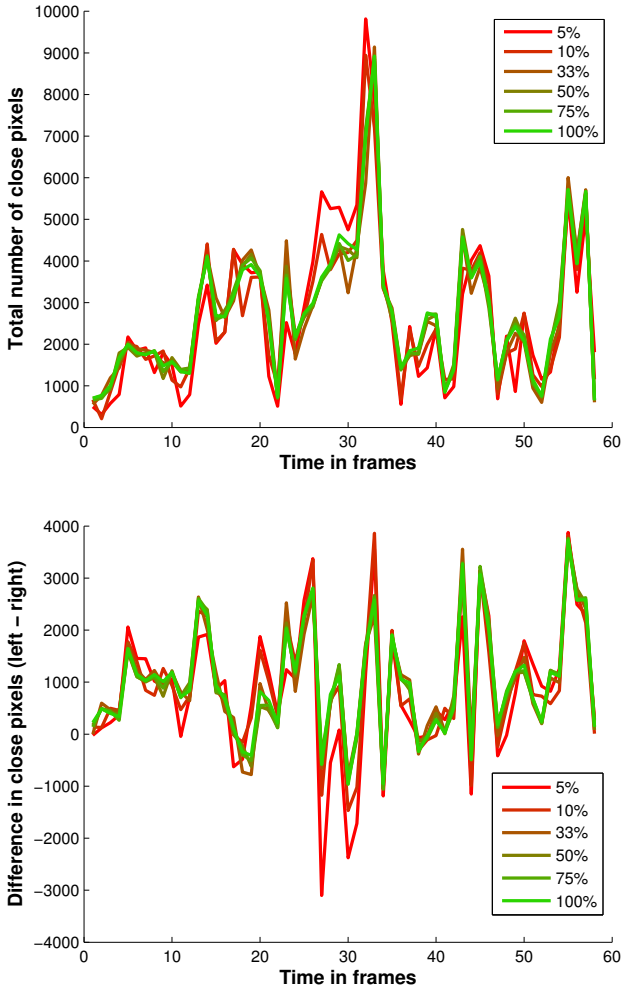

Figure 3.4: Effect of subsampling on the aggregate values used by the obstacle avoidance control algorithm. Top: total number of pixels with a disparity higher than 5 (close pixels). Bottom: difference between the number of close pixels in the left and right part of the image. The results are shown for various subsampling ratios, ranging from 5% (red) to 100% (green).

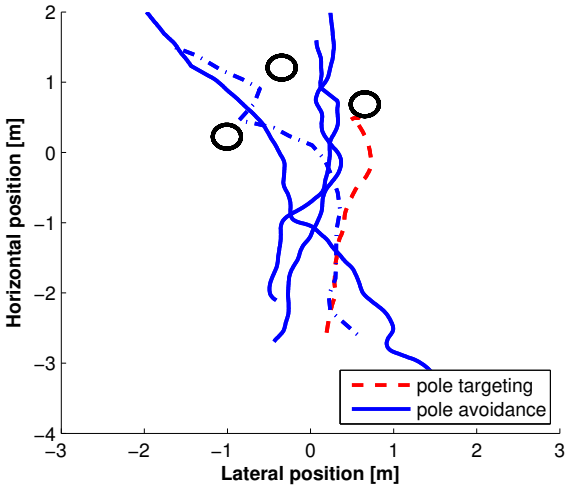

Figure 3.5: Flight trajectories of the DelFly Explorer in an area with three round vertical poles (black circles), shown from above. The obstacle avoidance trajectories are shown in blue. Three tracks show successful trials (solid blue). One track shows a trial where the DelfLy hit two poles with its wings (dash-dotted blue). Finally, one trajectory is shown where the control input was inversed in order to fly toward poles (red dashed).

3.4.4. Flight test experiment using short-term planning

In order to test the DelFly Explorer in more difficult environments, the avoidance algorithm proposed in Sec. 2.5.3 was also implemented. This resulted in autonomous obstacle avoidance with flight times up to 9 minutes in different environments. Videos of these tests and from the tests described in this study can be found online[2]. Note that the method from Sec. 2.5.3 is able to handle other important situations such as flying toward a straight wall or a corner.

3.5. Conclusions

The first lightweight flapping wing MAV flying autonomously with onboard stereo vision processing is demonstrated. Having the stereo vision processing on board has been made possible by: (1) the lightweight electronics for autopilot and stereo vision system, (2) the design improvements regarding the motor and ailerons, and (3) the development of a robust, computation and memory efficient, line-based stereo vision algorithm, named LongSeq. In particular, the quality of the disparity maps created by the stereo vision algorithm shows that it copes well with low visual texture (typical for indoor environments) and image deteriorations such as blur. The computational efficiency is enhanced with the help of sub-sampling, at a negligible cost in accuracy. The functioning of the system in the presence of flapping motion has been illustrated with an application to a sparse obstacle avoidance task,

[2] http://www.delfly.nl

including autonomous take-off and height control.

References

[1] A. Bachrach, R. He, and N. Roy, *Autonomous flight in unstructured and unknown indoor environments,* in *EMAV, the Netherlands* (2009).

[2] S. Grzonka, G. Grisetti, and W. Burgard, *Towards a navigation system for autonomous indoor flying,* in *(ICRA 2009), Kobe, Japan* (2009).

[3] A. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, *Visual odometry and mapping for autonomous flight using an rgb-d camera.* in *ISRR* (2011).

[4] M. Fallon, H. Johannsson, and J. Leonard, *Efficient scene simulation for robust monte carlo localization using an rgb-d camera,* (2012) pp. 1663–1670.

[5] K. Schmid, M. Suppa, and D. Burschka, *Towards autonomous mav exploration in cluttered indoor and outdoor environments,* in *RSS* (2013).

[6] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, *Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor,* in *RSS* (2013).

[7] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, *Controlled Flight of a Biologically Inspired, Insect-Scale Robot.* Science **340**, 603 (2013).

[8] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov, *Development of the Nano Hummingbird: A Tailless flapping wing micro air vehicle.* in *50th AIAA Aerospace Science Meeting* (2012) pp. 6–12.

[9] W. Shyy, H. Aono, C. Kang, and H. Liu, *An Introduction to Flapping Wing Aerodynamics* (Cambridge University Press, 2013).

[10] G. C. H. E. de Croon, K. M. E. de Clerq, R. Ruijsink, B. D. W. Remes, and C. de Wagter, *Design, aerodynamics, and vision-based control of the DelFly,* International Journal on Micro Air Vehicles **1**, 71 (2009).

[11] C.-L. Chen and F.-Y. Hsiao, *Attitude acquisition using stereo-vision methodology,* in *IASTED Conference* (2009).

[12] S. S. Baek and R. S. Fearing, *Flight forces and altitude regulation of 12 gram I-Bird,* in *IEEE RAS and EMBS Int Conf on Biomedical Robotics and Biomechatronics (BioRob)* (2010) pp. 454–460.

[13] F. Hsiao, H. Hsu, C. Chen, L. Yang, and J. F. Shen, *Using stereo vision to acquire the flight information of flapping-wing mavs.* Journal of Applied Science and Engineering **15**, 213 (2012).

[14] G. de Croon, C. de Wagter, B. Remes, and R. Ruijsink, *Random sampling for indoor flight,* in *International Micro Air Vehicle conference, Braunschweig, Germany (2010)* (2010).

[15] 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 **28**, 529 (2012).

[16] S. Tijmons, G. de Croon, B. Remes, C. D. Wagter, R. Ruijsink, E.-J. V. Kampen, and Q. Chu, *Stereo vision based obstacle avoidance on flapping wing mavs,* in *EuroGNC* (2013).

[17] S. S. Baek, F. G. Bermudez, and R. Fearing, *Flight control for target seeking by 13 gram ornithopter,* in *IEEE/RSJ Int Conf on Intelligent Robots and Systems.* (2011).

[18] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[19] F. G. Bermudez and R. Fearing, *Optical flow on a flapping wing robot,* in *IROS 2009* (2009) pp. 5027–5032.

[20] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, and A. Schmitt, *Real-time stereo by using dynamic programming,* in *Computer Vision and Pattern Recognition Workshop(CVPRW)* (2004) pp. 29–29.

[21] H. Hirschmuller, *Accurate and efficient stereo processing by semi-global matching and mutual information,* in *Computer Vision and Pattern Recognition (CVPR)*, Vol. 2 (2005) pp. 807–814.

[22] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, *Real-time plane-sweeping stereo with multiple sweeping directions,* in *Computer Vision and Pattern Recognition (CVPR)* (2007) pp. 1–8.

[23] G. de Croon, C. D. Wagter, B. Remes, and R. Ruijsink, *Sub-sampling: real-time vision for micro air vehicles,* Robotics and Autonomous Systems **60**, 167.

4

The Droplet Strategy: an Efficient Method for Obstacle Avoidance

The DelFly Explorer, as presented in the previous chapter, is a flapping wing MAV capable of performing obstacle detection using its onboard stereo vision system. Initial tests showed that pure reactive avoidance strategies lead to avoidance failures, while a strategy that plans a small distance ahead results in several successful flight of several minutes.

In this chapter, this avoidance strategy is further developed and extensively analyzed. A new efficient stereo vision algorithm is presented that improves the detection performance of the system. Both computer simulations and real-world flight tests are presented to show the robustness and effectiveness of the system in combination with the proposed avoidance strategy.

This chapter is based on the following article:
S. Tijmons, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, M. Mulder, *Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV*, IEEE Transactions on Robotics, (2017)

4.1. Introduction

Autonomous flight of Micro Air Vehicles (MAVs) is an active area of research in robotics. Because of its wide scale of potential applications it is gaining a growing amount of attention from governments and industry. Especially for outdoor applications, such as surveillance, monitoring, aerial photography and mapping, many commercial MAV systems are currently available. For indoor applications, however, this is not so much the case as these systems require more advanced methods for localization and navigation. This forms a challenge as more onboard sensors are required while indoor applications often require small sizes for the platforms. For lightweight MAVs under 50 g many solutions for autonomous navigation from the literature are therefore not applicable. Active sensors such as laser range finders [1–3] and RGB-D cameras [4] are typically heavier than the 50 g platforms themselves. Cameras, which are passive, are commonly used in combination with Simultaneous Localization and Mapping (SLAM) methods [5–7], dense reconstruction methods [8] or visual odometry methods [9] which provide information needed for navigation. A downside of these methods is their high demand for processing power and memory.

Obstacle avoidance and also other aspects of indoor navigation have nonetheless been demonstrated on several lightweight platforms. In most of these studies, with platforms ranging from 10 g to 30 g, onboard sensing is realized by using optical flow sensors [10–16]. These sensors can be very light and fast, but at the cost of providing low resolution. As a consequence, flow inputs generated by small obstacles are filtered out [11]. Optical flow also has the limiting properties that no flow information is available around the focus of expansion (direction of motion) [13], and that the flow estimates only provide relative speed information. As a consequence mainly reactive methods have been applied in these studies, that use the differences between measurements from multiple sensors to balance the distances to surrounding objects. This provides effective methods for specific guidance and avoidance tasks, but does not guarantee collision-free flight.

Stereo vision is considered to be a more robust method for the purpose of obstacle avoidance. The main advantages are that objects can also be detected in the direction of motion and that it provides absolute distance estimates to these objects. These advantages were demonstrated on a relatively heavy platform equipped with optical flow and stereo vision systems [17]. Onboard stereo vision has also been demonstrated on a fixed-wing vehicle of over 500 g flying at 9 m/s [18]. In flight tests small obstacles right ahead of the vehicle are detected at a range of 5 m and at a frame rate of 120 Hz. This approach shows that stereo vision can be used to combine short-term path planning with reactive avoidance control.

The main contributions of this study are the introduction of a computationally efficient avoidance strategy, its validation by simulation experiments, and its implementation and validation on an extremely light flapping wing MAV. The avoidance strategy ensures obstacle avoidance, even in complex and closed environments, based on information from a stereo vision system. The strategy is specifically designed for narrow and extremely lightweight systems, flying in narrow and cluttered environments that are restricted to, or prefer to maintain, a minimum forward ve-

locity while having a limited turn rate (nonholonomic constraint). The strategy explicitly takes into account the limited field-of-view of the cameras. The method does not require creating a map or storing camera images or disparity maps. This combination of properties results in an efficient and effective method for obstacle avoidance that is suitable for implementation on tiny, embedded systems. This is validated by computer simulations and real flight experiments.

A preliminary version of the avoidance strategy was tested in Chapter 2 on a flapping wing MAV that relied on off-board processing. The flapping wing MAV in the current study, which includes onboard processing, demonstrated a standard method for reactive avoidance in Section 3. The current study presents a number of important improvements compared to previous work. First, the "Droplet" strategy is introduced, which incorporates a new set of decision rules that improves its robustness, by taking into account the limitations of the onboard stereo vision system. In addition, the theory behind the strategy is described in detail, a theoretical proof of guarantee is presented and its computational efficiency is compared to related approaches. A different stereo vision algorithm running on board of the vehicle is presented that improves robustness, reliability and efficiency. The proposed avoidance strategy is evaluated through extensive computer simulations to show the effectiveness of the method in order to compare it with other reactive methods that have comparable computational complexity, and to analyze its performance in combination with the actual obstacle detection sensor. Furthermore, the avoidance strategy is demonstrated experimentally through test flights with the DelFly Explorer, a 20 g flapping wing MAV with onboard stereo vision processing. This is the first study showing obstacle avoidance based on an onboard stereo vision system with a real platform with such a low weight.

This chapter is organized as follows. Section 4.2 discusses related work on other flapping wing MAVs and different obstacle avoidance methods. In Section 4.3 the DelFly Explorer and its onboard stereo vision system are described. In Section 4.4 the avoidance strategy is explained. The avoidance strategy is compared to other methods and analyzed through simulations in Section 5.5. Flight experiments with the real platform and vision system are evaluated in Section 5.6. Finally, conclusions are drawn in Section 4.7.

4.2. Related work

The number of studies on autonomous capabilities of flapping wing MAVs is fairly limited, as much research focuses on the design of such vehicles. Their lightweight designs limit the possibilities to use onboard sensors, and many studies demonstrate guidance and control capabilities using ground-based tracking systems [19–24]. However, a number of studies also demonstrate control, guidance and navigation tasks on flapping wing vehicles. IMU-based attitude stabilization has been demonstrated on a 19 g platform [25] and attitude control on a 0.1 g platform [26, 27]. The latter also demonstrated height control using an optical flow sensor [28]. Guidance tasks have been realized on 16 g platforms, such as target-seeking, using an onboard Wii-mote infrared camera [29], and line following, using an onboard camera and an off-board processing unit [23]. Autonomous navigation-related tasks

demonstrated so far are vision-based obstacle avoidance indoors (but using off-board processing) with a 16 g platform [30] and GPS/IMU-based loitering outdoors with a 312 g platform [31]. So far, autonomous obstacle avoidance has not been demonstrated onboard a flapping wing MAV.

On other types of lightweight platforms a common approach is to use reactive control based on optical flow sensors [10–16]. Several studies on reactive methods mention the possibility of collisions with obstacles outside the field-of-view of the sensors [12],[32]. On heavier platforms either heavier active sensors such as laser rangefinders are used [1–4], or computationally demanding vision methods are applied, such as SLAM [6, 7], dense reconstruction [8] or visual odometry [9], using monocular or stereo cameras. These methods provide relative or absolute ranges to points in the environment which form the basis for localization and obstacle detection. This information enables path planning, which can be more robust than reactive methods, but at the cost of more computational load. A closed-loop rapidly exploring random tree (RRT) approach has been demonstrated in combination with stereo vision and SLAM on a 1 kg quadrotor [33]. This approach uses an efficient algorithm to check for collisions in disparity space while generating trajectories to candidate waypoints. Another study efficiently represents obstacle locations and possible vehicle states by using an octree-based state lattice [34]. This keeps memory consumption low and makes their A^* graph search for the optimal trajectory more efficient. The method is demonstrated on board a quadrotor equipped with a stereo vision system.

The latter study uses motion primitives, an efficient method for generating candidate trajectories that relies on a set of pre-computed control input sequences and trajectories which are checked for collisions. This has also been demonstrated on a real quadrotor equipped with a LIDAR sensor using a graph search algorithm [35]. A similar approach for path planning has been proposed for very small MAVs [36] but this study does not show experiments with a real perception sensor. Another study that also focuses on using motion primitives on very small platforms makes use of a receding horizon control-based motion planner, but again no real perception sensor was used in the experiments [37]. Their approach uses two types of motion primitives: steady turns for waypoint connection and transient maneuvers for instantaneous heading reversal. This combination of possible actions ensures the availability of an escape route, which is a shortcoming of reactive methods. Instead of storing a map with all detected obstacles, it has been proposed to use a local map of surrounding obstacles in combination with a fading memory, which remembers obstacle locations for a while, and deletes them later on [38]. This restricts memory requirements while allowing a planner to take into account obstacles outside the field-of-view.

This study proposes an avoidance strategy that guarantees obstacle avoidance and the availability of an escape route for a vehicle with nonholonomic constraints. The method does not require to store disparity information or a map with obstacle locations. The strategy is particularly beneficial in terms of robustness for systems that rely on small embedded perception sensors, such as the stereo vision system used in this study, as such sensors have various limitations: low resolution, sensi-

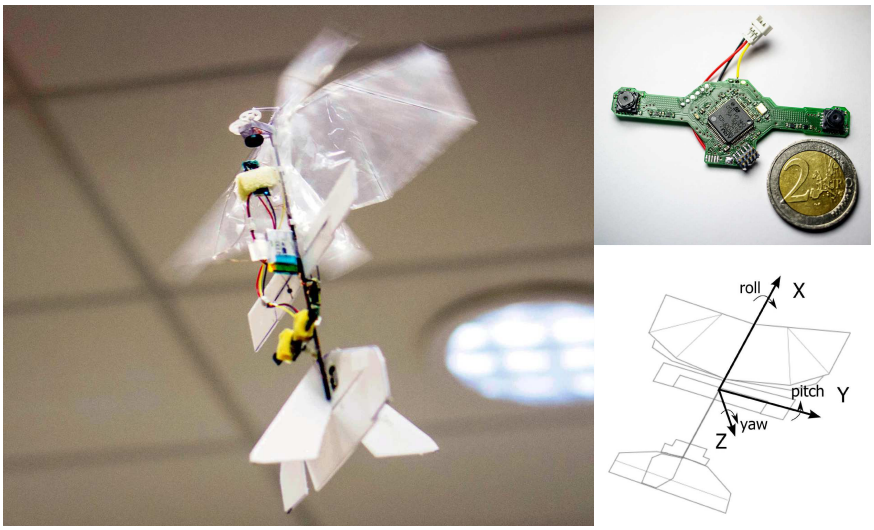

Figure 4.1: The DelFly Explorer platform. **Left**: The DelFly Explorer in slow forward flight condition. For more design details, see Fig. 3.1. **Top-right**: Custom-made stereo vision camera, baseline 6 cm. The coin illustrates its small size. **Bottom-right**: Body-axes definition.

tivity to motion (image blur), limited field-of-view, limited range. Furthermore the proposed avoidance strategy relies on an efficient algorithm that checks for collisions in disparity space and uses a small set of rules within a finite-state machine to make turn decisions.

4.3. System design overview

The avoidance strategy proposed in this study is tested on the DelFly Explorer, the Flapping Wing MAV shown in Fig. 4.1. This platform has a wingspan of 28 cm and a weight of 20 g. The Explorer version of the DelFly has a few important differences compared to its design used in previous studies ([30, 39], Chapter 2). First, it has a custom-made 1.0 g autopilot board which includes all electronics required for flight control: an electronic speed controller for its brushless motor, a transceiver for two-way communication with a remote station, and an ATmega328P 20 MHz microcontroller for onboard processing. The microcontroller has access to an MPU9150 9-axis IMU (gyro-, accelero-, magnetometers) and a BMP180 pressure sensor, which are used for attitude and altitude control.

Second, actuated aileron surfaces are added which create aerodynamic moments about the body X-axis (see Fig. 4.1). The ailerons are more effective for horizontal heading control in slow forward flight than the original tail rudder which acts on the vehicle's Z-axis.

Third, the vehicle is equipped with a custom-made stereo vision system of 4 g which is used for obstacle detection in this study. This sensor is further described in Sec. 4.3.2.

4.3.1. DelFly Explorer flight characteristics

The DelFly Explorer has two important differences compared to fixed-wing aircraft: the wings generate not only lift but also thrust, and the location of the center of gravity is further aft, close to the trailing edges of the wings. This enables the vehicle to fly passively stable at low forward speeds and at high pitch angles, as shown in Fig. 4.1. The forward flight velocity is typically 0.5-1.0 m/s and the pitch angle in the range of 70-80°. Forward velocity and pitch angle are controlled by the elevator and the motor speed. When the vehicle hovers or flies backward it is not passively stable.

The vehicle possesses a nonholonomic constraint in that it has no authority over its lateral velocity along the Y-axis. In the horizontal plane, it can only be guided by controlling the heading angle, for which the ailerons are used. The lateral velocity is damped by the surfaces of the wings and the vertical tail. Lateral drift is therefore determined by the velocity of the air. The vertical velocity is mainly controlled by the motor speed which determines the flapping frequency.

The lift forces produced by the wing and tail surfaces improve the flight efficiency considerably. Flight times of over 9 minutes have been recorded in this study at a forward speed of 0.6 m/s and using a Li-Po battery of 180 mAh. When hovering, the flight time reduces to around 3 minutes. This property forms an advantage of a flapping wing design over the more conventional quadrotor design. For example, the 19 g Crazyflie Nano Quadcopter[1] can fly for up to 7 minutes on a 170 mAh Li-Po battery, and this reduces to 3.5 minutes when a 6 g camera system is added [40]. To benefit from the higher flight efficiency, the avoidance strategy proposed here guides the DelFly such that it can maintain its forward velocity while avoiding collisions. At the same time, the vehicle also benefits from the passive stability characteristics when flying forward.

4.3.2. Stereo vision system implementation

The custom-made vision system on board the Explorer is shown in Fig. 4.1. It features two 30 Hz CMOS (TCM8230MD) cameras with 640×480 pixels resolution and a field-of-view of 58° × 45°. An STM32F405 microprocessor performs the stereo image processing. Memory restrictions (192 kB RAM) and processing restrictions (168 MHz) constrain the stereo vision algorithm to only use a sub-resolution of 128×96 pixels for calculating disparity maps. Disparity maps contain disparity (inverse distance) values per image pixel and thereby indicate the presence and location of obstacles.

To ensure computational efficiency, a sparse stereo vision method is implemented. The method is based on standard Sum of Absolute Differences (SAD) window matching [41]. For efficiency reasons, window matching is only tried at image locations where the horizontal image gradient (determined by horizontal differential convolution over single lines) contains a local peak and if this gradient exceeds a predefined threshold. For robustness reasons the output from the window matching computation is evaluated using a peak ratio test (see [42] for a comparison with other confidence metrics). Only if the ratio between the cost of the best

[1] https://www.bitcraze.io/

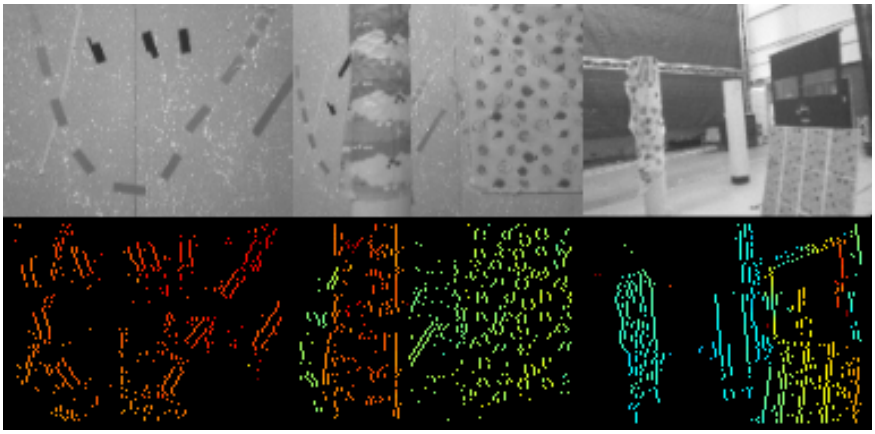

Figure 4.2: Example disparity images obtained from the stereo vision system using the sparse block matching method as used in this study. Black pixels are shown when no matches are found for these pixels. Red pixels represent high disparity values (small distances) and blue pixels represent low disparity values (large distances). Yellow and green represent medium values for disparity. The figure is best viewed in digital format.

match and the second-best match (excluding direct neighbors of the best match) exceeds a predefined threshold, the match is regarded reliable. Sub-disparity estimates are then calculated using parabola fitting using the three matching costs around the best match. The method runs at a frame rate of 15-30 Hz. This rate varies with the amount of texture observed in the image. Note that at least some form of texture is required to detect an obstacle.

The stereo vision algorithm differs from the method presented in Sec. 3.3.1. The algorithm in the previous study is intended to deliver dense disparity maps in an efficient way, and is specifically tuned to provide sensible disparity estimates in image regions that lack texture (e.g., smooth walls). However, the resulting disparity maps contain a lot of bad matches due to the assumption of fronto-parallel planes and also contain a relatively high degree of noise. The method produces a lot of noise specifically in image regions that contain dense texture, which is undesirable. Furthermore the quality of the estimated disparity values cannot be monitored and can vary considerably within a single image. The algorithm implemented in the current study produces only sparse disparity maps but solves the main issues of the previous method: it returns disparity estimates only for points with relatively high certainty, the certainty is higher in texture-rich image regions and the number of bad matches is significantly reduced. As a result, sparse disparity maps are produced that contain disparity values with a relatively high certainty and accuracy, while the number of computed disparity points can be used as a measure of reliability. Furthermore, the average frame rate is also higher compared to the previous method ($>$15 Hz compared to $\sim$11 Hz).

Fig. 4.2 shows examples of the sparse disparity maps computed by the stereo vision system. These examples illustrate how much depth information is obtained

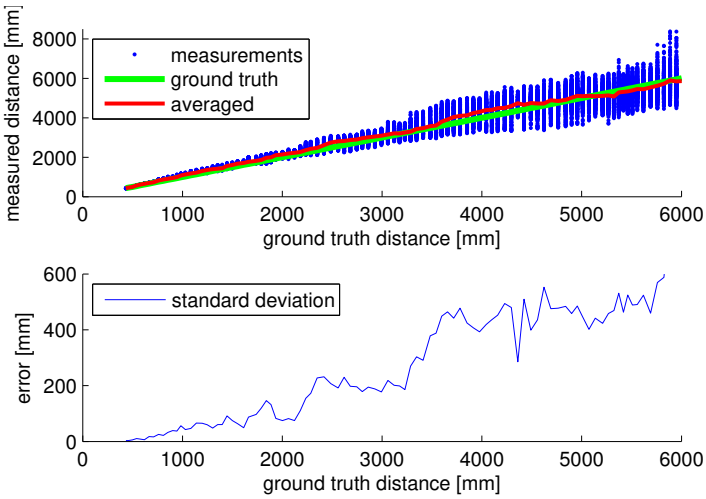

Figure 4.3: Performance characteristics of the camera system shown in Fig. 4.1. Estimates are based on 150-300 stereo matches per frame. **Top**: Spread of estimated distances from individual stereo matches, as well as averaged estimated distance. **Bottom**: Standard deviation of the distance error.

and how this relates to the quantity of image texture. The examples also serve as a way to show that the quality of depth information is more important than its quantity. The most right image serves as a good example. The left part of this image contains a sparse amount of information. Yet it can be assumed that no near obstacles are present on this side of the image, while on the right side of the image near obstacles are present.

Characteristics of the camera system performance are shown in Fig. 4.3. These results give an insight in the accuracy with which the system can detect distances to obstacles, and how this accuracy declines with increasing distance. Note that at a distance of 3 meters, the standard deviation of the estimated distance is around 200 mm. This observation is important as this distance corresponds to the maximum that needs to be observed in the experimental flight tests.

4.4. Description of the avoidance strategy

4.4.1. The proposed avoidance maneuver

The starting point of the proposed avoidance strategy is that, when avoiding obstacles, the vehicle performs steady turns with a constant flight speed and a constant turn rate. By maintaining its forward speed the vehicle benefits from a higher flight efficiency, better stability and sufficient response to aileron control inputs, as explained in Section 4.3.1. This all means that the vehicle will perform avoidance maneuvers with a constant turn radius R_{turn} given by:

$$R_{turn} = V/\dot{\psi} \tag{4.1}$$

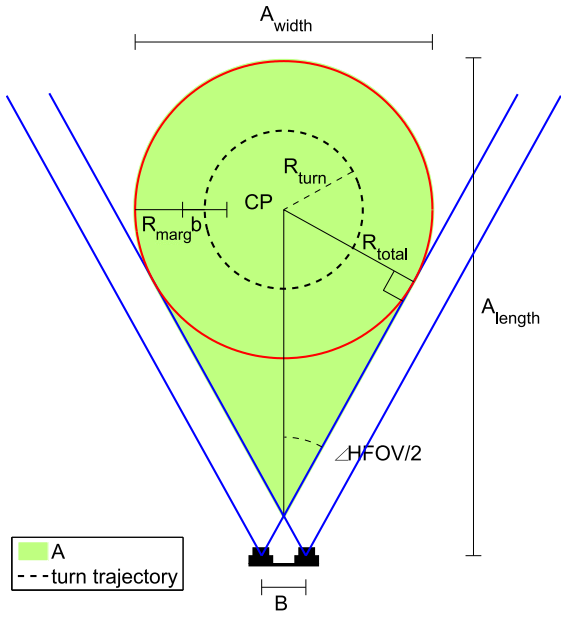

Figure 4.4: Top-view of the Droplet avoidance area showing its contours and parameters that define its shape. The filled (green) area defines the region within the field-of-view of the cameras (indicated by the black symbols at the bottom) that needs to stay clear of obstacles. The shape of this area is defined such that a vehicle is able to fly circles within this region (as indicated by the dashed line). The Droplet region is defined in the camera reference frame.

Both forward speed V and heading turn rate $\dot{\psi}$ primarily depend on vehicle dynamics. Section 4.5.2 will show, however, that also the update rate of the stereo vision system affects the range of possible velocities and turn rates that can be set. Fixed values are assumed for the forward speed, turn rate and turn radius. To avoid collisions, the vehicle needs a sufficiently large circular turn space with radius:

$$R_{total} = R_{turn} + b/2 + R_{marg} \tag{4.2}$$

In this equation b is the vehicle wingspan and R_{marg} is an error margin to account for deviations from the avoidance trajectory and inaccuracies in measured distances to detected obstacles.

The novelty of the method presented here is that it continuously checks if there is such a turn space ahead of the vehicle that is free of obstacles. This is illustrated in Fig. 4.4 by a top-view schematic. The turn space with radius R_{total} is indicated by the red circle with center point CP. The position of CP is at some distance ahead of the camera such that the turn space fits within the horizontal field-of-view $HFOV$ of the cameras, which is defined as the overlapping part of the two blue cones. The green area A then defines the minimum area that needs to be free of obstacles in order to guarantee a safe avoidance maneuver. Because of the shape of this area, this method is called the "Droplet" strategy. Avoidance maneuvers are

initiated when obstacles are detected inside this area.

In contrast to many other avoidance maneuvers found in the literature, the proposed maneuver does not only guarantee collision avoidance up to the end of some proposed path. In fact, this method proposes an infinite path as it makes sure that the vehicle can fly the circular turn trajectory indicated in Fig. 4.4.

The need for finding a free space ahead of the vehicle, stems from the fact that the cameras have a limited field-of-view. This issue can be tackled by using multiple camera systems pointing in all directions or by using wide-angle/panoramic lenses. Instead of adding extra payload weight by using one of these approaches, the proposed method only requires the execution of a few additional control rules.

The distance CP_{dist} between the camera and center point CP is given by:

$$CP_{dist} = \frac{R_{total}}{\sin(HFOV/2)} + \frac{B/2}{\tan(HFOV/2)} \tag{4.3}$$

In this equation, B is the baseline of the camera system, which is 60 mm. $HFOV$, the horizontal field-of-view, is 58° for this system. The size of the Droplet area A is important as it defines the minimum size of spaces that can be accessed by the vehicle. The outer dimensions of this space are given by:

$$A_{width} = 2R_{total} \tag{4.4}$$

$$\begin{aligned} A_{length} &= CP_{dist} + R_{total} \\ &= R_{total}\left(1 + \frac{1}{\sin(HFOV/2)}\right) + \frac{B/2}{\tan(HFOV/2)} \end{aligned} \tag{4.5}$$

In these equations, A_{width} is the width of the area A, and A_{length} is the length of area A. The relations from Equations 4.4 and 4.5 are visualized in Fig. 4.5. The $HFOV$ values are typical for cameras with normal to wide-angle lenses. For the turn radius a range of values is shown that fits to the characteristics of the vehicle from this study. The forward speed is typically 0.5-1.0 m/s, while lower speeds are possible but not desirable for several reasons. At turn rates of 1-2 rad/s the value of R_{turn} would be in the range of 0.25-1.0 m which proved to be realistic numbers in real test flights. The wingspan b of the DelFly is 28 cm and the value of the error margin R_{marg} is set at 30 cm.

4.4.2. Obstacle detection method

The task of the stereo vision system is to detect whether there is any obstacle present within the Droplet area. Based on the shape of this region, a threshold disparity value can be calculated for each camera pixel such that the Droplet shape is defined in a reference disparity map. As the Droplet shape is static, this reference disparity map can be precomputed, making the obstacle detection process computationally inexpensive. By comparing each new observed disparity map with the reference disparity map, the number of pixels can be counted that exceeds the

reference value. If this number is higher than a threshold $\tau_{d>ref}$ (=7 px in the current experiments), it is assumed that an obstacle has entered the Droplet area.

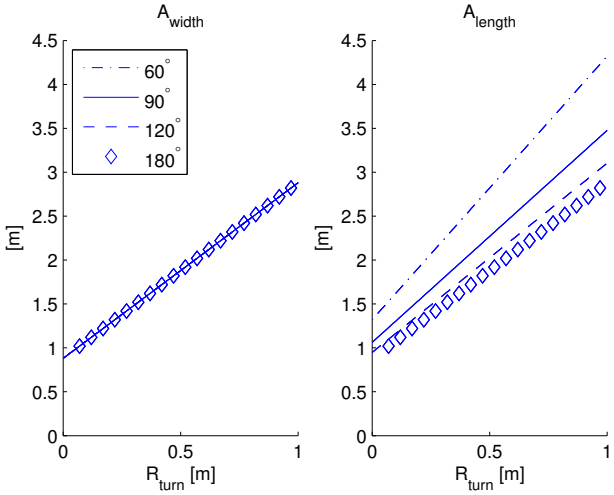

Figure 4.5: Dimensions of the Droplet avoidance area. These dimensions are dependent on the turn radius (R_{turn}) of the vehicle and the horizontal field-of-view of the camera ($HFOV$). The width A_{width} is shown at the left, which is independent of the size of $HFOV$. The length A_{length} is shown at the right.

This approach forms the bare implementation of the Droplet method. This implementation would be sufficient to perform obstacle avoidance in an ideal case (large, well-textured obstacles, perfect sensing), but leads to detection failures in many real world situations. To improve robustness in those cases, the bare detection method is extended in two ways.

First, the sparsity of the disparity maps is used as a quantitative measure of image texture. The stereo vision algorithm produces only disparity values for pixels at image locations with high intensity gradients. Therefore, the number of such pixels is counted in the left and right halve of the disparity image. If either of the two halves contains a number of such pixels that is lower than a minimum threshold $\tau_{\#d_{\min}}$(=50 px), the algorithm treats this disparity image similarly as if an obstacle would present in the Droplet area. This rule deals with texture-poor surfaces (i.e., white walls). The disparity images can also be split in more segments and in multiple ways, but splitting it in two halves turned out to be very effective according to experiments.

The second adjustment deals with situations where objects are hard to observe, for instance if only the edge of an obstacle is visible. This occurs when objects do not contain sufficient texture or if their appearance resembles the appearance of the background. In those cases it is possible that only a small part of the obstacle is observable in the disparity image, or that it is not consistently observable in a series of frames. The probability of detecting that obstacle can be improved by using disparity information from a series of subsequent frames. The main idea is that pure

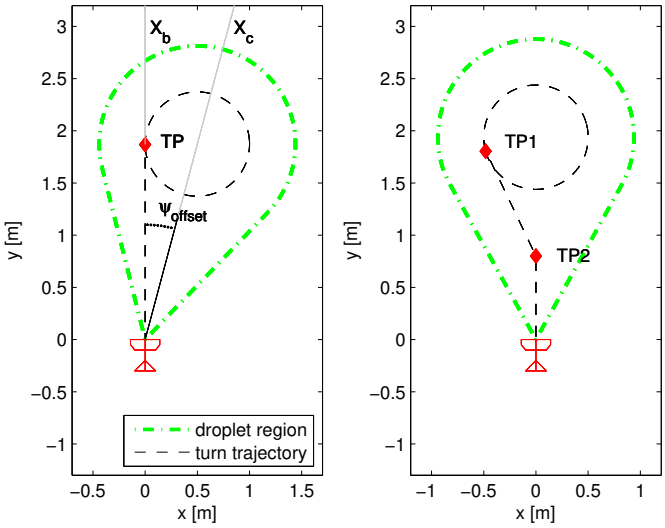

Figure 4.6: **Left:** Visualization showing how the Droplet strategy is implemented on the vehicle. The dashed (black) line shows the trajectory of the vehicle once the avoidance maneuver is initiated. The camera offset angle ψ_{offset} shows how the camera is mounted such that the Droplet region (indicated by the dash-dotted/green line) encloses the avoidance trajectory. The turn point TP indicates the point where the vehicle starts turning if the avoidance sequence is initiated. Note that the trajectory is defined in the body reference frame of the vehicle. **Right:** Alternative implementation of the Droplet strategy where the camera system is aligned with the body axis of the vehicle. This requires an extra turn point ($TP2$) where the vehicle steers towards $TP1$.

noise will result in observations containing random disparity values, while sparse observations resulting from a real obstacle will contain consistent disparity values that increase over time. Therefore a detection-vector is maintained that represents a 1-D grid with size of 0.1 meter, where each grid cell functions as a counter. For each observed disparity value that indicates that an object is present within the Droplet area, it is calculated at what distance the vehicle should initiate the avoidance maneuver (explained in next section). This distance is then matched with the detection-vector, and the corresponding counter is increased by one. Each time step, the distances are reduced by assuming a constant velocity - this means that the first elements in the detection-vector are deleted. New disparity occurrences are added to previous ones and can be regarded as probabilities that there is an actual obstacle at that distance. When a turn is initiated, the vector is reset.

4.4.3. Implementation of detection and avoidance methods

Based on the safety region as defined by the Droplet region in Fig. 4.4, simple control rules are formulated to perform robust obstacle avoidance. This is further clarified by Fig. 4.6, which shows the trajectory the vehicle will fly once an obstacle is detected (black dashed line), starting at the location indicated in the figure. Note

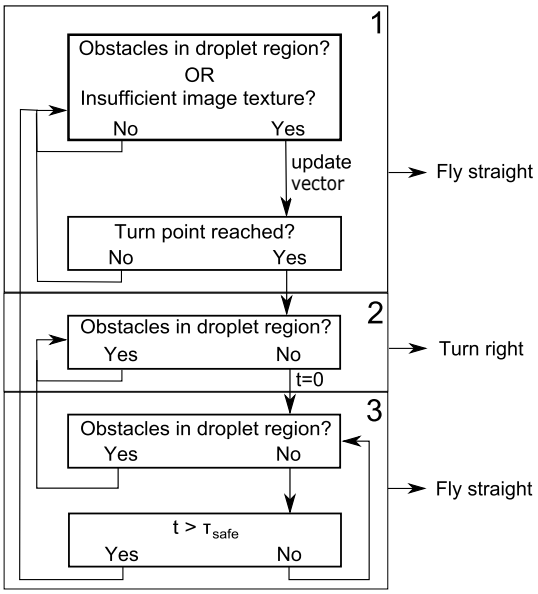

Figure 4.7: The state machine for controlling the vehicle. The states define which outputs/actions are required, and include tests to check if a new state needs to be activated, or not. The value of τ_{TP} is precomputed, the value of τ_{safe} is a fixed tuning parameter.

that in this figure the coordinate system is defined with reference to the vehicle and that the camera system is mounted on the vehicle with a heading offset angle ψ_{offset}. This offset angle is defined as:

$$\psi_{offset} = \arcsin\left(\frac{R_{turn}}{CP_{dist}}\right) \tag{4.6}$$

By introducing the offset, the avoidance maneuver can be performed in two steps, rather than three. First, the vehicle maintains its heading and flies to the turn point TP. Second, when TP has been reached, the vehicle will perform a turn with constant rate and speed. This two-step procedure is visualized by the (black) dashed line in Fig. 4.6 (left). It is assumed that the course of the vehicle is exactly the same as the heading angle. This assumption only holds if there is no crosswind and if the lateral drift is small. Also note that the vehicle performs only right turns. If ψ_{offset} would be the same but negative, the vehicle would have to perform left turns.

If ψ_{offset} would be zero or would have a different value, the heading angle of the vehicle would not be aligned with the direction of the turn point. An additional turn would then be required where the vehicle aligns with the circular path. This is shown in Fig. 4.6 (right). Thus ψ_{offset} serves to reduce the complexity of the avoidance maneuver.

The time τ_{TP} needed to reach TP is given by:

$$\tau_{TP} = \frac{TP}{V} = \frac{\sqrt{CP_{dist}^2 - R_{turn}^2}}{V} \tag{4.7}$$

Note that this timing value is only valid in case objects are detected as soon as they enter the Droplet area (perfect sensing). As explained in Sec. 4.4.2, two adjustments were made to the detection algorithm to make it more robust in the real world. First, it is checked if there is sufficient information present in the stereo images. If this is not the case (not sufficient disparity values), such observation is regarded unreliable and the system should respond similar as to the case when an obstacle is detected. The timing value of Eq. 4.7 is used in these cases to determine when the vehicle should start avoiding. The second adjustment takes into account distances to detected obstacles. For each point detected in the Droplet area it is computed how far it has penetrated this area. In other words, the distance between each detected point and the upper border of the Droplet area is computed. These individual distance estimates are used to obtain updated location estimates of TP. These corrected distance estimates are used for updating the obstacle-detection vector. By checking at each time step whether the value of the obstacle-detection counter that corresponds to the current vehicle location exceeds threshold $\tau_{d>ref}$, it is determined if a turn point has been reached. If that is the case, an avoidance maneuver is initiated.

The diagram in Fig. 4.7 shows the finite-state machine which is designed to ensure that the vehicle will always remain within the safety region. The first state will be active for as long as no avoidance action is required. Once the obstacle-detection vector indicates that a turn point has been reached (threshold $\tau_{d>ref}$ is exceeded), the second state activates. In this state the vehicle will perform the steady turn. The turn will continue until the vision system detects a new heading for which an obstacle-free Droplet region is found. In that case, the third state becomes active, and the vehicle will fly straight again. If the vision system detects obstacles while this state is still active, the second state immediately becomes active again and the vehicle is instructed to continue turning. If no obstacles are detected for a predefined amount of time (defined by the threshold τ_{safe} =1 s) the system will return to the first state.

The reason for adding the third state is twofold. First, due to inertia the heading angle response will have some overshoot when the vehicle stops turning. New obstacles might then be present on the right side of the Droplet region. Second, turning leads to more motion blur in the camera images. Some obstacles are therefore only properly detected while flying straight. The third state increases the chance of detecting obstacles. It can potentially be left out if the onboard control system would take care of overshoots and when more sensitive cameras would be used that suffer less from motion blur.

It is emphasized that all parameters that can be set are fixed during flight. This holds for the vehicle dynamics, such as forward speed and turn rate, and with that all other settings that define the Droplet shape. All aspects of the avoidance strategy are therefore fully precomputed which makes the algorithm extremely efficient.

4.4.4. Theoretical guarantee of collision-free flight

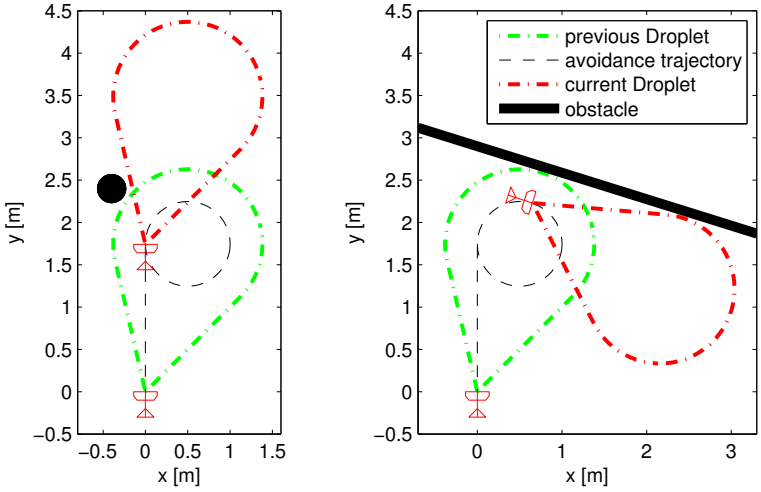

Figure 4.8: Illustration of the working principle of the Droplet strategy.
Left: Example of the avoidance of a small round obstacle. In this specific case, the avoidance maneuver is triggered when the obstacle is detected within the lower droplet area (green). Once the vehicle reaches the turn point, the droplet area (red) is found to be free of obstacles, and the avoidance maneuver is aborted. **Right**: A more general example of the avoidance of a wall. In this case, the avoidance maneuver is triggered when the wall is detected within the lower droplet area (green). The vehicle will turn according to the predefined avoidance trajectory until the droplet area (red) is found to be free of obstacles.

The droplet strategy is guaranteed to avoid collisions, given perfect sensing and actuation. The main reason for this is that a robot employing the strategy will always move within free space that it has observed before, which is illustrated by examples in Fig. 4.8. There are two conditions to this guarantee: (1) there is no obstacle to the front left of the robot at initialization, and (2) the margin R_{marg} is large enough. The reason for the first condition is illustrated by Fig. 4.9. The yellow (solid) triangle indicates a region right in front of the vehicle that is not covered by the field-of-view of the camera system. At initialization there should be no obstacle in this region.

The second condition ensures that for the remainder of the flight this same (yellow) region fits within an earlier observed free droplet region. This is true if R_{marg} is of sufficient size. As defined in Fig. 4.9, the size of L_1 can be determined as:

$$L_1 = \frac{b}{2\tan\beta} = \frac{b}{2\tan(HFOV/2 - \psi_{offset})} \tag{4.8}$$

See Fig. 4.9 for a definition of β. Furthermore, R_{total} can then be expressed as:

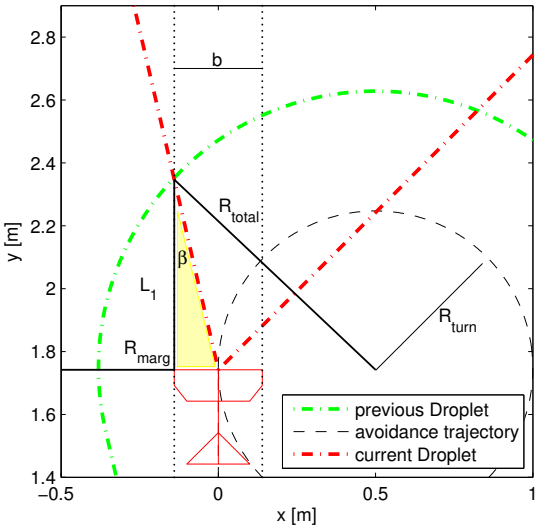

Figure 4.9: Visualization showing that the vehicle always flies in free spaces that it has observed before. In the current situation the vehicle is enclosed by a previously observed Droplet region (green). It will be enclosed by the currently observed Droplet region (red) as soon as it leaves the previous (green) region along a tangent line of the turn circle.

$$R_{total} = \sqrt{L_1^2 + (b/2 + R_{turn})^2} \tag{4.9}$$

Using Eq. 4.2, a minimum value for R_{marg} can then be obtained. This is the theoretical minimum value of R_{marg}, assuming no errors. If the actual value of R_{marg} is larger than R_{marg}^{min}, the Droplet strategy guarantees collision-free flight. Since in the real world, sensing and actuation are not perfect, it is better to take a higher margin. Indeed, the Droplet size from Fig. 6 has R_{marg} =30 cm, which is larger than the theoretically required R_{marg}^{min} of 21 cm.

4.4.5. Extending the avoidance maneuver to 3D

The Droplet maneuver can be extended to 3D by taking into account the ability of the vehicle to climb and descend. The vertical speed of the vehicle can be controlled independently from the horizontal states and is determined by the motor speed, which regulates the flapping frequency. Fig. 4.10 shows the droplet area (green) both from the top as well as from the side. From the side-view it can be seen that the camera vertical field-of-view ($VFOV$) plays a role in how this region is defined vertically. The height of this area is restricted to A_{height} which is defined as:

$$A_{height} = h + 2H_{marg} \tag{4.10}$$

In this equation, h is the vertical size of the vehicle and H_{marg} is an error margin to account for obstacle detection inaccuracies and for altitude variations. The latter is determined by the performance of the height control loop. The red box indicates the region that must be kept clear of obstacles to allow a safe horizontal avoidance maneuver. It corresponds to the red circle in the top-view; its width is equal to the radius of the red circle.

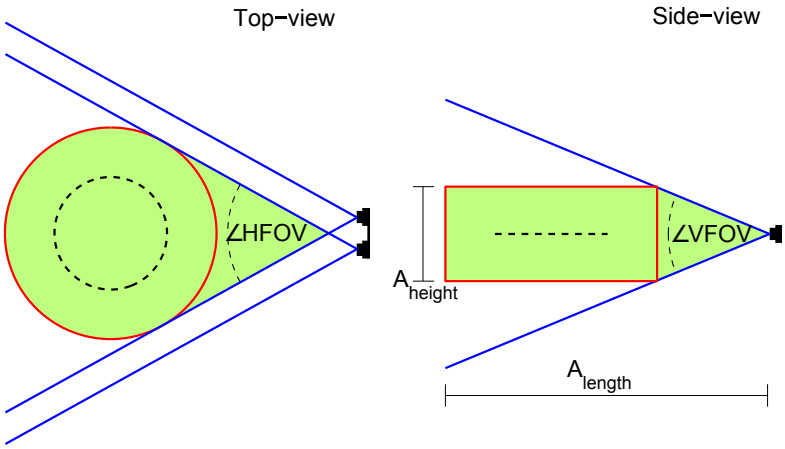

Figure 4.10: Top view and side view of the droplet region. The green region needs to be checked for obstacles. The turn region, indicated by the red lines is defined as a disk with height A_{height}. for clarity the horizontal ($HVOF$) and vertical ($VFOV$) field-of-view angles of the cameras are indicated.

The regions above and below the red turn region (side-view) are observed by the cameras as well. For initiating vertical maneuvers these regions need to be checked for obstacles. By doing so, it can be guaranteed that after a vertical maneuver the corresponding turn region is shifted into a region without obstacles. A possible method to implement vertical maneuvers is to extend the length of the droplet area, as shown in Fig. 4.11 (top-left). Here it is shown how the turn region can be moved upwards (black rectangle) to allow a climb. However, this method requires the length of the droplet area to be increased substantially (dashed lines), while the inaccuracy of the stereo vision system degrades significantly at larger distances.

The other diagrams of Fig. 4.11 show a more compact method. The top-right diagram shows that the length of the droplet region does not need to be increased. The climb area (blue) and descent area (red) are indicated which fill up the remainder of the $VFOV$. By stitching observations in these regions together over time, these areas are stretched out. The middle-left diagram shows how the observed area has grown while the camera has moved to the indicated location. This is a specific location; at this location the corresponding turn region (indicated by the dashed red box) is the same region as observed earlier in the starting situation as

shown in the top-right diagram. The middle-right diagram illustrates that a climb maneuver can be followed from this point on. By following a climb path parallel to the border of the $VFOV$ it is guaranteed that the corresponding turn region will always be enclosed within the region observed previously.

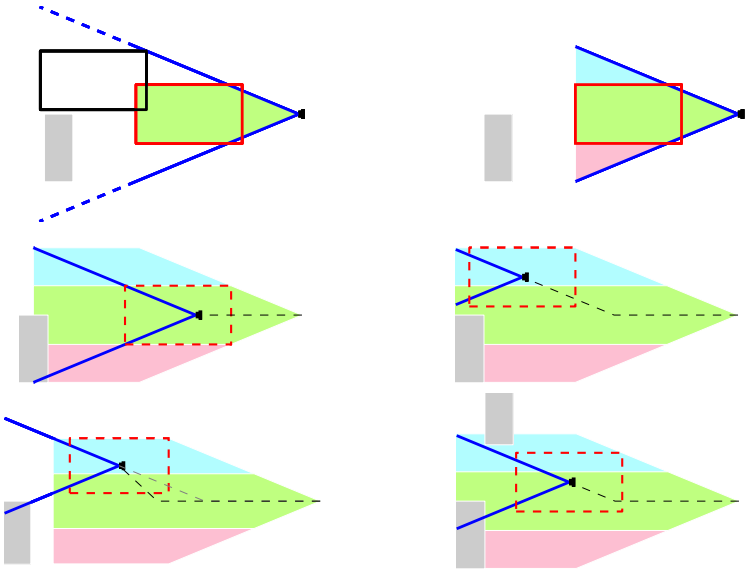

Figure 4.11: Possible implementations of height control into the Droplet avoidance strategy. The gray boxes indicate obstacles that need to be avoided. **Top-left**:length of the droplet is extended such that the future turn region (black box) fits when shifted vertically. **Top-right**: length of the droplet is not changed, but observations from the areas above and below the droplet area are stitched together. After checking for obstacles over a certain amount of distance (**middle-left**) a climb maneuver can be initiated. If an obstacle is then detected within the Droplet area (green), the climb is executed (**middle-right**). If the obstacle is further away (**bottom-left**), the climb will be initiated later, as soon as the obstacle is detected. The climb can be steeper in this case. If an obstacle is detected in the climb region (blue), the climb can only be executed for as long as the corresponding turn region allows.

The method is further defined and generalized in Fig. 4.12. First note that in this example configuration, the red turn region is not fully enclosed by the $VFOV$. The size of the non-enclosed area is indicated as D_{misfit}, which is defined as:

$$D_{misfit} = A_{length} - \frac{A_{height}}{2 \cdot \tan(VFOV/2)} \tag{4.11}$$

$VFOV$, the vertical field-of-view is $45°$ for the camera system in this study. Depending on the sizes of A_{length} and A_{height}, the value of D_{misfit} can also be zero or negative, meaning that the whole area is enclosed in the $VFOV$. D_{misfit} is important as it determines at which point a vertical maneuver may be initiated. For clarity, the vertical maneuver point VMP is defined. It is different from the turn point TP, which is used for the horizontal maneuvers. These points are related by:

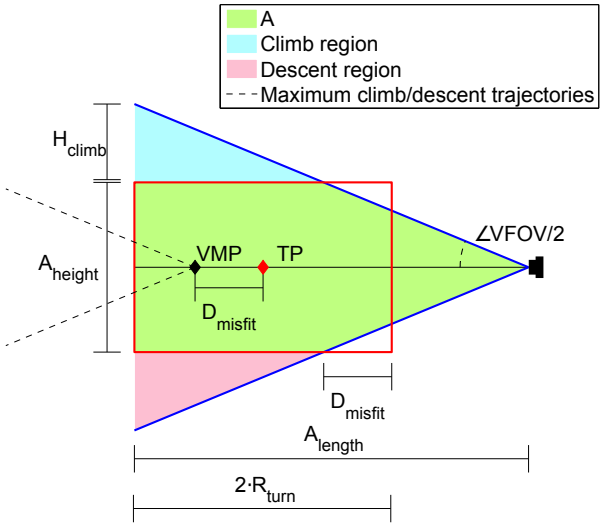

Figure 4.12: Side-view of the Droplet avoidance area. The filled inner (green) area defines the region within the field-of-view of the cameras (indicated by the black symbol at the right) that needs to stay clear of obstacles. The length of this area, A_{length}, corresponds to the definition of Fig. 4.4. The filled outer (blue) areas define regions that need to be clear of obstacles to allow a climb or descent maneuver. Such a maneuver is only safe to execute after the vehicle has travelled at least up to the vertical maneuver point VMP. The maximum rate for a climb/descent is limited by the field-of-view of the camera and the maximum climb/descent distance is defined by the field-of-view and A_{length}.

$$VMP = TP + D_{misfit} \tag{4.12}$$

In the configuration as presented in Fig. 4.12, VMP is farther away than TP. The additional shift D_{misfit} ensures that the turn region (red box) corresponding to TP is shifted into the field-of-view before the vehicle starts climbing.

After reaching VMP, a vertical maneuver will be executed as soon as an obstacle is detected within the Droplet area. If this happens immediately at VMP, the maximum climb/descent rate is limited by the slope of the $VFOV$ borders. This is shown in the middle-right diagram in Fig. 4.11. If an obstacle is detected later after reaching VMP, the vertical maneuver is also initiated later, and a higher climb/descent rate is allowed. This is shown in the bottom-left diagram of the same figure. The maximum climb/descent height difference H_{climb}, indicated in Fig. 4.12, follows from the sizes of the $VFOV$ and the turn region. It is defined as:

$$H_{climb} = A_{length} \cdot \tan(VFOV/2) - A_{height}/2 \tag{4.13}$$

Just like the horizontal avoidance maneuvers, the vertical maneuvers need to be planned ahead, based on the observations from the camera system. To incorporate height control, disparity observations need to be compared with two reference disparity maps. The first reference disparity map only represents the Droplet area

(green), and is also used for the horizontal control decisions. The second reference map represents the whole $VFOV$ up to a distance A_{length}, thus also the spaces above and below this area (blue and red). This combination allows to identify whether obstacles are detected within the Droplet region or in the climb/descent regions. The finite-state machine from Fig. 4.7 can be maintained to incorporate height control. For as long as the system is in the first state, it needs to be checked if the climb and/or descent regions are free of obstacles. No obstacles should be detected in these areas while travelling for a distance D_{misfit} first. As long as this is not the case, the location of VMP cannot be fixed, meaning that a vertical maneuver is not safe. If the climb or descent region is found to be free of obstacles for a distance D_{misfit}, the location of VMP is fixed according to the definition in Fig. 4.12. Thereafter, the distance travelled without detecting any obstacles determines how far the vehicle may climb or descent when it reaches VMP, and also over which distance. The climb angle is defined by the slope of the $VFOV$. Recall that the maximum climb/descent height is limited as Eq. 4.13 shows. As Fig. 4.11 illustrates, a vertical maneuver is only initiated if an obstacle (gray box) is detected within the Droplet region after reaching VMP (middle-right and bottom-left diagram). The bottom-right diagram of this figure shows the situation where an obstacle is present in the climb region. In such cases, the maximum climb/descent height is smaller than the limit specified by Eq. 4.13.

4.4.6. Comparison of computational complexity with the state-of-the-art

As stated in the introduction, the proposed avoidance strategy is computationally efficient. To put this claim into perspective, the computational and memory complexity of the proposed method is compared to related methods. Different elements of such methods can be distinguished: sensors (with or without dedicated processor), post-processing steps (changing sensor data representation, e.g., making a map) and control algorithms (e.g., reactive, path planning, the Droplet method). In analogy to Section 4.2 most of the methods can be categorised as either reactive methods or path planning methods. Reactive methods are computationally extremely efficient, as they typically just compare some sensor values to pre-set thresholds, with a few if/else statements [14] or a small neural network [12]. However, as was mentioned, and as will be demonstrated in Section 5.5 by simulation experiments, reactive methods do not guarantee collision avoidance. As explained in Section 4.4.1 the Droplet strategy requires a straightforward comparison between disparity maps generated by the stereo vision system and a precomputed reference disparity map. This is the only input for the simple state machine. The computational complexity of this approach is therefore in the same order as reactive methods.

As this study focuses on an obstacle avoidance strategy embedded into a real MAV, its computational complexity is compared to three path planning approaches that are intended for applicability to MAVs. The first approach it is compared with uses stereo vision in combination with an RRT planner [33]. It requires three steps to compute a safe path: an expansion-operation on the disparity map to correct

for the size of the MAV, the computation of dynamically feasible trajectories to randomly proposed waypoints using a closed-loop RRT algorithm, and a check whether candidate trajectories collide with the obstacles detected in the disparity map. The computational complexity is not specified in detail but it is mentioned that on a 1.86 GHz processor this method produces motion plans at 2Hz. Furthermore it should be noted that the disparity maps are generated by a separate stereo camera system with a dedicated processor. The second approach it is compared with also uses a separate stereo vision system to produce disparity maps [34]. This information is converted into a memory efficient octree-based search lattice. An A^* graph search is used to find a collision-free optimal path to a goal state. For generating the motion plans based on the disparity map a 1.7 GHz processor is used. It uses under 30% of CPU and about 400 MB of memory to deliver motion plans at approximately 2 Hz. The third approach it is compared with uses a LIDAR to detect obstacles [35]. Using these measurements an occupancy grid is obtained which is used by the motion planner (variant of A^*) to find feasible trajectories based on motion primitives, taking into account the 3-D footprint of the vehicle. For their experiments on the real platform a 2 GHz processor is used. Using 60% of CPU a motion plan is computed at 0.5 Hz on average. For comparison, it is noted once more that the obstacle avoidance system proposed in this study combines all steps from sensing to control decisions on a 4 g vision system that relies on a single 168 MHz processor with 192 kB memory, and still runs faster than 15 Hz. Hence, the memory required is more than 2000 times smaller than the approach of [34] (400 MB / 192 kb = 2142) and – by taking the processor speed for the same update rate and assuming 1 core used – at least 75 times faster (1.7GHz / 168 MHz $\approx$ 10, 15 / 2 = 7.5). This is a very prudent estimate, since the numbers of the droplet strategy include the stereo vision processing time and memory, while those of [36] exclude the stereo processing information.

It is further noted that the three studies that use path planning also rely on additional sensors for pose estimation and localisation, either by running an onboard SLAM algorithm or by relying on external tracking. On the other hand, path planning methods perform a more sophisticated task than pure obstacle avoidance; they aim to arrive at a goal position. A downside of path planning methods in general is that accurate information of obstacle sizes and locations is required to compute safe paths around them. The method proposed in this study circumvents this need, which makes it suitable for small scale systems with embedded sensing and processing.

4.5. Simulation experiments

A simulation setup was created in *SmartUAV*, an in-house developed software environment, to compare the proposed avoidance strategy with other reactive strategies from the literature and to analyze effects of several parameters on its performance. The software simulates the motion of the vehicle, the visual inputs to the stereo vision camera, the stereo vision algorithm (described in Section 4.3.2) and the Droplet control loop. The motion of the vehicle is simulated at 50 Hz, the vision and control loops are simulated at 10 Hz.

Because the performance of the avoidance strategy is the point of interest in the simulations, two external factors of influence are ignored. First, wind disturbances are not simulated because the platform currently does not have the ability to estimate this. Second, the simulations are performed in a highly textured environment such that the performance of the stereo vision algorithm is reliable and constant. These simplifications allow for a good comparison of different avoidance strategies.

In the setup, the DelFly is flying in a square room of 6×6×3 meter with textured walls as shown in Fig. 4.13. Every single run, five obstacles, having the same texture as the walls (white in the figure for visibility), are randomly placed to increase the difficulty of the avoidance task. Note that this is a very challenging environment, since it is a relatively small, closed space with additional obstacles. The obstacles are vertical poles with a diameter of 40 cm and a height of 3 m. It is assured that the DelFly starts in a position where it will detect an obstacle-free Droplet region. Each experiment run ends as soon as a crash occurs or stops after 600 seconds of uneventful flight. This time limit represents the maximum flight time with a single battery. The DelFly flies at a constant height of 1.5 m and with a constant velocity of 0.55 m/s.

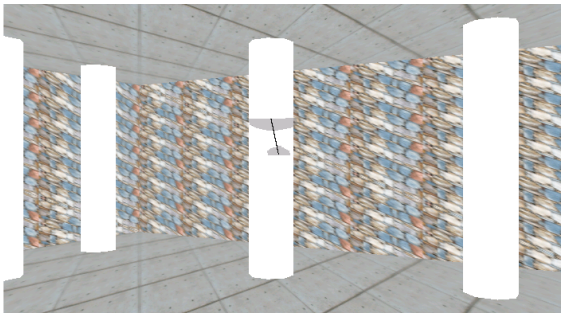

Figure 4.13: Screenshot of the SmartUAV simulator that shows a DelFly model flying in a simulated room. In this example the walls are highly textured, the vertical poles are white and have no texture (for visibility of this image), the floor and ceiling are visualized as concrete stones.

4.5.1. Comparison with purely reactive methods

The Droplet strategy is compared to two other strategies. The first one is based on the method proposed by [43] and [10] which aims to balance the average optical flow as measured by cameras on the left- and right-hand side of the vehicle. Since the vision system in this study obtains a single disparity map instead of optical flow from two different cameras, the disparity map is split into a left and right half, and the average disparity values of the two halves serve as the input that needs to be balanced. The method often resulted in crashes in one of the corners, since the walls act as a 'funnel' in these cases. A constant turn rate offset (18% of maximum turn speed 120 °/s) is therefore added which results in successful flight when the room is free of obstacles. The offset is chosen such that the resulting behavior is comparable to the results described in [43]. It was further tuned during simulations

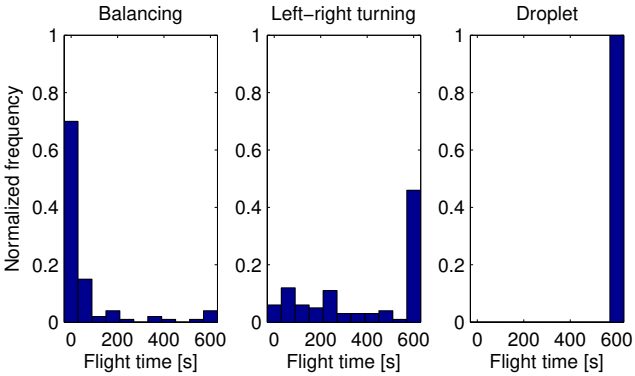

Figure 4.14: Comparison of performance of three different reactive avoidance methods. The methods are indicated as Balancing method [43], left-right turning method [30] and Droplet method proposed here. The histograms visualize flight time until an obstacle is hit, with a maximum of 600 seconds. The results are normalized and are based on 100 runs for each method.

to obtain the best performance results for the eventual flights with obstacles. This method is referred to as the *balancing* method.

The second strategy is based on a method proposed in de Croon $et\ al.$ [30]. This method is based on time-to-contact estimates obtained from optical flow. Based on such estimates from the left and right halves of the camera images, it is determined whether the vehicle should turn, and in which direction. Once a turn is initiated, it is continued for a fixed amount of time to avoid oscillations. In this study a disparity threshold (4 px) is defined for both halves of the disparity map. If sufficient pixels ($\geq$10) in one of the halves of the disparity map exceed the threshold, the vehicle starts turning with maximum turn rate (120 °/s) in the direction opposite from the obstacle. In this implementation a new turn direction can only be chosen once the vehicle flies straight again. This method is referred to as the *left-right turning* method.

Results of the comparison are shown in Fig. 4.14 and 4.15. For this comparison, the Droplet strategy has the dimensions as shown in the top-left of Fig. 4.16. The value for R_{turn} is 263 mm, which follows from the selected velocity (0.55 m/s) and turn rate (120 °/s). Fig. 4.14 shows the distribution of flight times of 100 runs for the different strategies in separate normalized histograms. The first histogram indicates that the balancing strategy fails in most cases. For the implementation in the original study [43], the system was tuned for an empty room. Neither the vision system nor the control strategy was supposed to cope with other obstacles in the test room. The two images on the left in Fig. 4.15 show the tracks of a failed flight and a successful flight using this method. The successful flight demonstrates that a safe route is found for certain obstacle setups. The failed flight is more representative for the general behavior, however. It demonstrates that relatively small obstacles influence the flown trajectory but not sufficiently to steer the vehicle away from too narrow passages.

The left-right turning strategy has a much higher success rate. The two middle

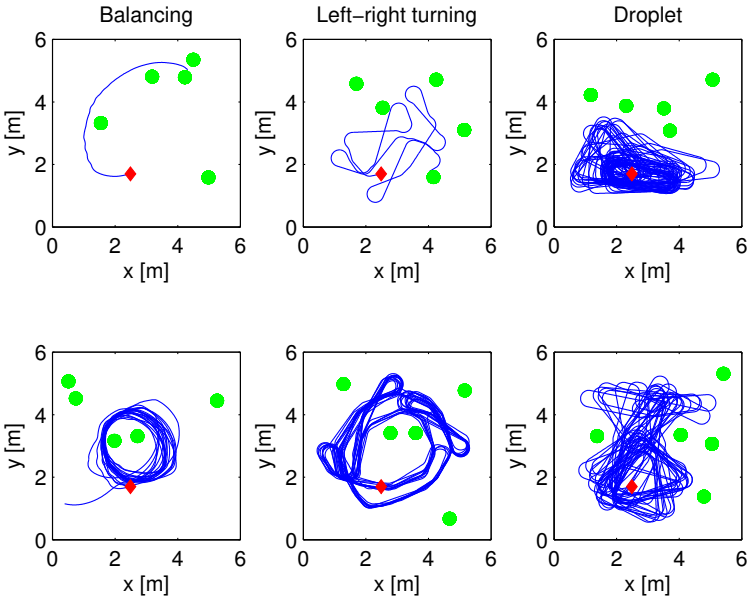

Figure 4.15: Example flight trajectories showing the behaviors of the three avoidance methods in a room of 6×6 meters containing five obstacles (indicated by the green circles). The red rhombus indicates the starting position. A failed flight (top-left) and a successful flight (bottom-left) of the Balancing method are shown. The two middle graphs show a failed flight (top) and a successful flight (bottom) of the left-right turning method. The two right graphs show two successful flights of the Droplet method proposed in this study. These trajectories show how the locations of the obstacles affect the areas covered by the vehicle.

plots in Fig. 4.15 show the tracks of a failed flight and a successful flight. The plot of the failed flight shows a typical failure case for this method. Due to the limited camera field-of-view, an avoidance turn is initiated in the direction of an unobserved obstacle. The track of the successful flight shows that this method allows the vehicle to reach a large part of the room. It can also be noted that the resulting flight trajectories are repetitive, which is typical.

Fig. 4.14 shows that the method proposed in this study results in a 100% success rate, as expected. The two plots on the right in Fig. 4.15 illustrate that the region of the room covered during the flight strongly depends on the obstacle locations. It can also be observed that the flight tracks are less repetitive when compared to the other strategies.

4.5.2. Analysis of the effects of parameter variations

In this section the influence of the shape of the Droplet area on the flight trajectories is analyzed. As a baseline, the same values as in the simulations from the previous section are used: a forward velocity of 0.55 m/s and a turn rate of 120 °/s. Again,

the Droplet shape is defined as shown in Fig. 4.16(a). Three different cases of parameter variation are analyzed.

Higher forward speed and turn rate

When the turn speed and turn rate are increased by the same factor (1.2), the shape of the avoidance region does not change. The time to reach the turn point τ_{TP} will decrease because the avoidance maneuver is flown at a higher speed of 0.66 m/s. Since the turn rate is increased but the stereo vision update rate is not changed (fixed at 10 Hz for all simulations), the heading sampling angle between the stereo vision measurements during the turn increases. As a result, the vehicle will stay longer in turns as the chance of finding a safe heading is reduced. Fig. 4.16(d) shows an example where the vehicle is locked at a few turn locations for long periods. This result illustrates that for selecting the values of forward speed and turn rate, the update rate of the vision system is also important.

Lower forward speed

By lowering the forward speed (0.36 m/s) while keeping the turn rate the same, the turn radius becomes smaller. This results in a decrease of the width, length and area of the avoidance region, which is visualized in Fig. 4.16(e). As a result the vehicle is able to access smaller spaces and the coverage should increase. Fig. 4.16(f) shows a flight trajectory which is an example of a flight where the vehicle is indeed able to cover multiple areas in the room.

Larger field-of-view

If only the horizontal field-of-view is increased (from 60° to 90°), the width of the avoidance region does not change, but the length and area are decreased as shown in Fig. 4.16(g). This should also have the effect that smaller spaces can be reached and that the coverage increases. Fig. 4.16(h) shows a flight for this configuration illustrating that the vehicle is able to reach different parts of the room.

Table 4.1 shows statistics for the varying parameter configurations based on 200 runs per parameter setting. These results show that flying at higher speeds and turn rates has the effect that the vehicle will be turning more often. Note that the number of turns is almost the same, but that the mean time per turn is increased. It can also be seen that this has a negative effect on the total coverage. Using the bootstrap method [44] it is determined that this decrease is significant (p<0.01).

Flying at a lower speed (but with the same turn rate) reduces the size of the Droplet region but leads to an increase in the area covered during a flight. This increase is also found to be significant with p<0.01. The amount of time spent on turning is decreased. On the other hand, the number of turns increases. As a result, the time per turn is decreased on average. These facts indicate that by lowering the flight speed, the vehicle makes more but shorter turns and is able to reach more places.

Increasing the horizontal field-of-view angle does not lead to a significant increase in area covered ($p = 0.32$). Apparently the width of the avoidance region is a more crucial factor than the size of the field-of-view. The mean time spent on turning reduces ($p = 0.05$) and the total number of turns increases. This can

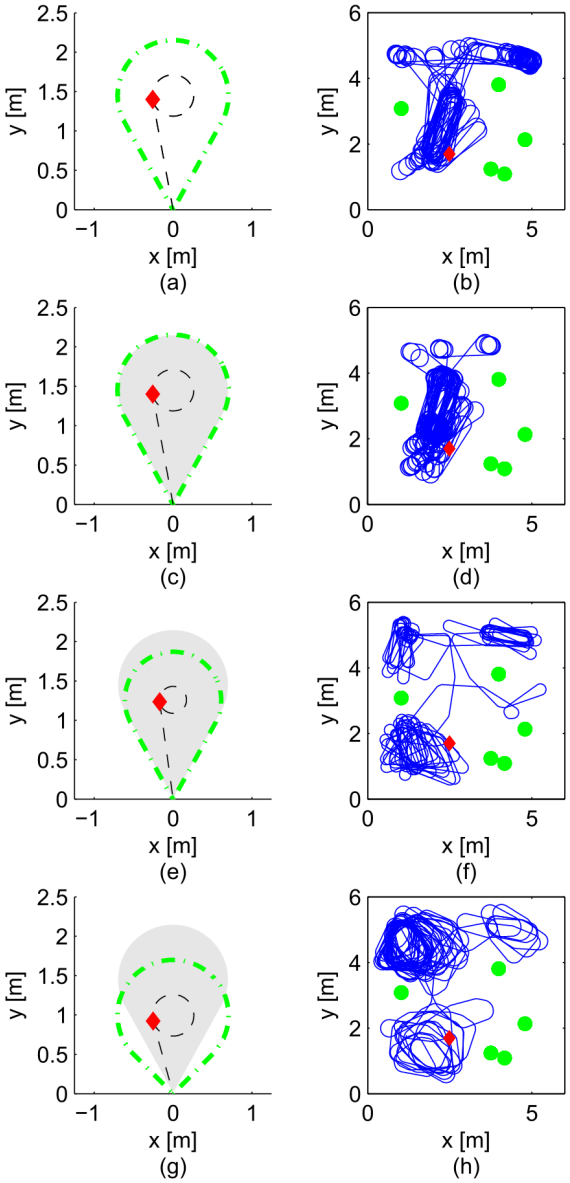

Figure 4.16: Shape of the Droplet region for different parameter settings (left column) and example trajectories for these Droplet shapes (right column). They gray regions indicate the shape as defined in Plot **a**, which is the shape of the baseline configuration. Plots **c-d** correspond to the case of higher forward speed and turn rate, Plots **e-f** are for lower forward speed and constant turn rate, and Plots **g-h** are for a larger field-of-view. The room layout is identical in the four cases to provide a clear visual comparison.

Table 4.1: Comparison of behavior of the Droplet method for different parameter settings

	Base	1)	2)	3)
Mean Coverage[1] (%)	52.3	44.9	58.0	53.8
Mean Time Turning (%)	38.2	49.4	24.4	35.1
Mean Number of Turns	118	117	138	270
Mean Time per Turn (%)	0.32	0.42	0.18	0.13

Base: Reference parameter setting for the Droplet region as defined in Fig. 4.6. **1)** Higher forward speed and turn rate. **2)** Lower forward speed. **3)** Larger camera field-of-view.

[1] Coverage is computed by dividing the test room in 36 patches of 1×1 meter and measuring how many patches are visited by the vehicle at least once during a flight. Mean coverage is computed as the average coverage over all flights.

be explained by the reduced distance to the turn point (CP_{dist}), which leads to a reduced total time needed to perform the Droplet maneuver. More separate avoidance maneuvers are performed with smaller turn angles, which reduces the mean time turning and the mean time per turn.

4.5.3. Comparison of obstacle detection rules

In Sec. 4.4.2, different obstacle detection rules are introduced. Two additional rules were described to deal with situations where objects might not be detected robustly. The effectiveness of these additional rules is demonstrated by experiments in which the texture of the walls and the poles in the simulated room are varied. The performance difference between the standard rule and the additional rules is compared in terms of success rate and room coverage. The results are listed in Table. 4.2. The left column indicates what is different compared to the fully textured room. Coverage is again expressed as percentage of visited area, averaged over all flights. The success rate indicates which percentage of 100 runs resulted in 600 seconds of collision-free flight.

An important observation is that the applied detection rules are very effective to prevent collisions with poorly textured surfaces, at the cost however of lowering the area covered by the vehicle. The success rates increase considerably for all conditions with poor texture. An interesting observation is that the covered area increases when the poles are not textured. Apparently, the vehicle visits locations that are never reached in case all obstacles are perfectly detected.

Table 4.2: Comparison of performance of obstacle detection rules

Rules:	Standard		Additional	
	Coverage (%)	Succes (%)	Coverage (%)	Succes (%)
Fully textured	46.5	100	37.8	100
One white wall	45.3	65.0	37.4	100
Two white walls	26.7	25.0	18.8	100
Four white walls	0	0	5.1	100
White poles	56.5	60.0	51.2	95.0

4.5.4. Simulation results of extension to 3D

To evaluate the advantage of implementing height control in the Droplet strategy, experiments with different pole configurations were performed. Changing the length of the poles and attaching them to either the floor or the ceiling requires the vehicle to fly over and under the poles. In the experiments the room has a height of 3 meter and the vehicle starts its flight at 1.5 meter altitude. Long poles extend from floor to ceiling, short poles have a length of 1.3 meter. Short poles that are attached to the ceiling thus start at a height of 1.7 meter. 100 flights of 600 seconds are performed for each configuration. The performance difference of the Droplet strategy with and without height control is compared in terms of total area covered. Furthermore the number of climb/descent actions is counted and the average flight altitude is computed.

100% success rates were achieved in all experiments. This shows the robustness of the proposed 3D implementation. More results are listed in Table. 4.3. By replacing the long poles with short poles, the average area covered during flights increases. Note that this number does not approach 100%. It could be further increased if specific rules are applied to enforce this. By applying different combinations of low poles and high poles, the average flight altitude changes accordingly. Note that the impact of the specific combination of low and high poles on the covered area is very small. Apparently the randomness of flight directions increases in this situation, leading to a higher coverage in any case.

4.6. Real-world Flight Experiments

Several flight tests with the real DelFly Explorer have been performed to evaluate the Droplet avoidance strategy. First, a set of eight flight tests was conducted that mimics the scenario from the simulation experiments. Furthermore, several flight tests were performed in different unadapted real world rooms to show the robustness of the method in all kinds of situations.

Table 4.3: Effectiveness of height control for the Droplet strategy

	Cov. [%]	# climbs.	#descents	altitude [m]
5 long poles	59.2	2.7	2.1	1.5
4 long 1 low poles	64.4	4.1	2.4	1.7
3 long 2 low poles	70.3	4.8	3.1	1.8
2 long 3 low poles	78.9	5.9	3.3	1.9
1 long 4 low poles	85.0	6.0	2.8	2.0
5 low poles	86.2	7.3	3.1	2.0
4 low 1 high pole	84.0	8.6	5.7	1.9
3 low 2 high poles	83.1	9.6	8.3	1.7
2 low 3 high poles	83.9	8.1	8.3	1.4
1 low 4 high poles	84.5	7.2	7.4	1.2
5 high poles	84.4	6.9	7.2	1.0

4.6.1. Experiments in simulator-like environment

A set of eight flight tests has been performed in a scenario that is comparable to the setup of the computer simulations. Fig. 4.17 (left) shows the test location where walls are placed to form a closed square room. The tests mimic the simulation tests from Section 5.5: the room measures 6×6 meters, it is well-textured and contains four highly-textured obstacles at varying locations. The location of the vehicle is tracked using an OptiTrack[2] motion capture system. The vehicle is fully autonomous during the test flights. The altitude is controlled using feedback from the pressure sensor, heading is controlled based on the data from the stereo vision camera. Note that in these experiments the obstacle detection adjustment rules are not implemented. Instead, good texture is assured for all objects. Roll rate and pitch rate feedback from the gyroscopes is used to stabilize the heading and pitch angle. The flight speed (≈0.6 m/s) is not regulated but set at the start of the flight by the trim position of the elevator. This elevator/speed setting allows for flight times of up to 9 minutes. A trim value for the motor rpm is set to minimize vertical speed. This trim value is set higher for the case a turn is made. Due to aileron deflections while turning, extra drag behind the wing is generated, which lowers the effective thrust. The higher setting for motor rpm mostly compensates for this loss during turns.

The flight speed and avoidance turn rate have been tuned to obtain a good avoidance performance during the test flights. The corresponding Droplet shape is as shown in Fig. 4.6 which has a length of 2.9 m and a width of 1.9 m. It is larger than the Droplet region indicated as baseline in the simulations (length of 2.1 m), because a slightly higher flight speed is set to have longer flight times. To

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

keep the turn rate the same, a larger turn radius is then needed. In fact, also a lower turn rate is selected, resulting in a turn radius (R_{turn}) of 0.5 m. The error margin (R_{marg}) is the same, 0.3 m. The lower turn rate is necessary to prevent the vehicle from staying in lengthy turns. Such a situation can occur if there is only a small margin available for fitting the droplet region in between the poles. Due to the combination of a high turn rate with a limited frame rate of the vision system, the chance of observing a free droplet region is fairly limited in some situations. This effect is reinforced by a rule in the second state of the state machine that checks if a free droplet region is observed in two consecutive frames. This rule is meant to improve robustness of the second state as the turning motion of the vehicle increases the blurring effect, which leads to more false-negatives of the stereo matching algorithm. An additional cause for the lengthy turns can be an overshoot in final heading angle when switching to the third state. As a result new obstacles might come into view, making the state machine to switch back to the second state.

Figure 4.17: **Left**: Test room of 6×6 meters where flight tests have been performed. The room contains four poles of 40 cm diameter that form obstacles. The locations of these obstacles were different for each test flight. **Middle and Right**: Images of two setups to test the robustness of the system to poorly textured obstacles. The orange poles do not have additional texture.
Middle: room with two walls, enclosed by the orange poles. **Right:** same room as in the left case, one pole is placed inside the enclosed area.

These considerations make clear that the choice for the parameters in Eq. 4.1, which are forward speed, heading turn rate and turn radius, are important to obtain robust performance. This is especially true for this test room which is relatively small and cluttered. A high forward velocity leads to long flight times and a small turn rate leads to better stereo vision results during turns. This leads to large values for the turn radius and thus to large droplet sizes. This is undesirable as the accuracy of the stereo vision system degrades with distance (worse obstacle detection) and because this also results in lengthy turns, as discussed previously. Long turning times have led to several collisions due to random drifting of the vehicle over time. The selection of the turn parameters from Eq. 4.1 turned out to be the most important step to obtain robust performance. Apart from this it was experienced that trimming and tuning the vehicle in between test flights is also crucial. The trim setting of the aileron is important to minimize the drift of the heading angle, which is necessary to fly straight when executing the first part of the droplet maneuver. Several crashes

occurred in cases where the vehicle would drift to the right during the first phase, resulting in crashes while executing the turn maneuver. The aileron trim setting also affects the induced drag caused by the aileron, which affects the response of the vehicle during a turn. Too wide turns were another reason for several crashes. Therefore, tuning of the aileron turn command is required to obtain the desired turn rate. Experience shows that by careful construction the aileron system can be made more robust, but will degrade after long flight times. Note that tuning the Droplet parameters only needs to be done once for a certain vehicle configuration. Trimming the aileron offset and the aileron turn command needs to be done also in between flights. Future work will focus on making the system more robust in this sense.

Robust performance is obtained given that the previously discussed conditions are satisfied. Furthermore, external factors such as the air flow and the lighting conditions play an important role in the robustness of the algorithm. Due to the relatively low wing loading of this specific vehicle it is important to have stable wind conditions, as drafts make the vehicle drift away. Several crashes occurred in cases where a sudden draft was experienced. Additional feedback on the position/velocity of the vehicle would be required to make the system more robust to this influence. This could be achieved by tracking surrounding objects using an additional algorithm in the stereo camera system, or by using an additional sensor. The quality of the disparity maps produced by the stereo camera system is obviously influenced by the lighting conditions since the camera sensors are passive. It was observed that daylight conditions resulted in a far more stable performance. By turning off artificial lighting or by testing without external sunlight, crashes would often be caused by a poor obstacle detection performance. Furthermore, it was mentioned previously that the turn rate of the vehicle has an important effect on the blurring of the camera images. The robustness of the method cannot be guaranteed when the camera system cannot produce reliable estimates. The limitations of the vision system might improve in the future as camera technology progresses.

The flight trajectory of Test 5 is shown in Fig. 4.18. Part of the trajectory is indicated by the dashed line. This part of the flight starts immediately after finishing a turn, which means that State 1 is active. Further on, States 2 and 3 become active, respectively. It can be observed that during this maneuver the DelFly keeps a safe distance to the obstacle it approaches. Other parts of the flight track show that during other approaches of this pole the vehicle gets much closer. This effect can be explained by the sensitivity of the DelFly to small gusts (non-constant flight speeds) and non-perfect performance of the stereo vision system.

Results from the test flights with different obstacle locations are shown in Table 4.4. Only successful flights are recorded that were achieved after obtaining good trim settings. The $Number\ of\ Turns$ is shown twice; the actual amounts of turns that were performed during the flights are shown, as well as the time-corrected numbers of turns, such that the test flights can be compared with the simulated flights of 600 seconds as shown in Table 4.1. Note that during flight Test 2 and 3, the DelFly performed over a hundred turns autonomously while flying in a 6×6 m space with several obstacles. The covered area in the real test flights

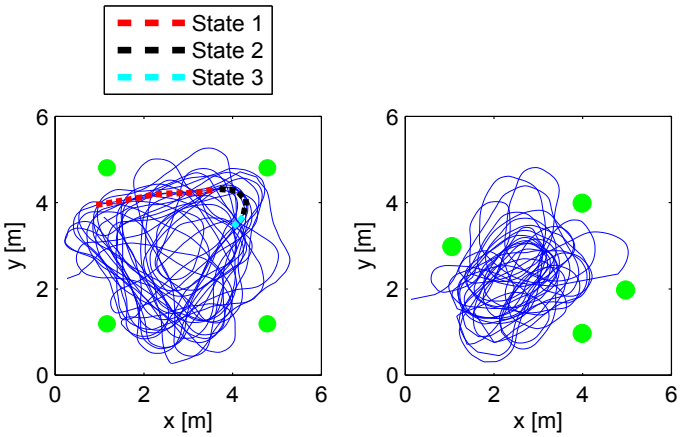

Figure 4.18: **Left**: Flight trajectory from Test 5. During this flight of 410 seconds the vehicle covered a distance of 271 meters. The locations of the obstacles (40 cm diameter) are indicated using the (green) circles. The dashed line shows one complete cycle of the Droplet strategy; the sub-track starts in State 1 (directly after a turn) and stops after completing the next turn.
Right: Flight trajectory from Test 8. During this flight of 295 seconds the vehicle covered a distance of 178 meters. The locations of the obstacles correspond closely to the obstacle locations in Fig. 4.16.

is higher which can be explained by the number of obstacles being four instead of five. Note that the $Time\ per\ Turn$ found for the flight tests with the longest durations (1-6, 8) matches the result found for the baseline case in Table 4.1. The large variation in total flight time can be partly explained by the trim setting of the aileron that was tuned at the start of each flight. A small difference in trim setting can have a significant effect on the drag it creates.

In Fig. 4.18, the flight trajectory of Test 8 is also shown. The obstacle locations in this test are very similar to Fig. 4.16 to compare the flight track from the real flight with the computer simulation. It can be observed that the lower part of the room is better covered during the real flight than in the simulation which is caused by the more random nature of the DelFly behavior in real life. The upper part of the room is not visited by the DelFly during the real flight test. This can be explained by the larger size of the Droplet region in the test flights.

4.6.2. Effect of extended obstacle detection rules

The results from the previous tests demonstrate the performance of the Droplet method in case the poles and the walls in the room are well-textured. To test the influence of the extended obstacle detection rules similar experiments are performed where no additional texture is added to the orange poles. As illustrated by Fig. 4.17, a small setup is created where a part of the original test room is used which is demarcated by the poles. In the first experiment (middle image) an open test space is created, in the second experiment (right image) one pole is placed inside this test space to increase the complexity.

Table 4.4: Flight trajectory results from real test flights

	1	2	3	4	5	6	7	8
Coverage (%)	69	64	75	64	72	64	58	61
Time Turning (%)	41	37	41	53	40	46	45	47
Nr. of Turns	85	119	122	90	94	72	38	76
*Nr. of Turns**	*127*	*130*	*136*	*175*	*138*	*128*	*89*	*154*
Time / Turn (%)	0.32	0.28	0.30	0.30	0.29	0.36	0.50	0.31
Flight time (s)	400	550	539	308	410	336	256	295
Trav. distance (m)	250	337	344	192	271	203	154	178
Avg. speed (m/s)	0.62	0.61	0.64	0.62	0.66	0.60	0.60	0.60

* The number of turns indicated is scaled to a total flight time of 600 seconds

To illustrate the complexity of the task and the challenge for the avoidance algorithm in this particular setup, Fig. 4.19 shows examples of stereo vision outputs. The figure illustrates that the distinctive orange color of the poles does not result in a distinctive intensity difference in comparison with the background. The same scene is shown for two conditions, one while the camera is static (left image), and one while the camera is carried on board the DelFly Explorer in flight (right image). These examples make clear that due to motion blur, the resulting disparity maps can be affected significantly. The proposed detection rules take into account that the number of points in the disparity maps is considerably reduced and varies more due to motion blur. In general a useful number of points is still detected, making the detection method very effective.

Results from flight test in the setups from Fig. 4.17 demonstrate that the system is able to robustly detect and avoid the orange poles when using the extended detection rules. Flights of 8 minutes and 4.5 minutes were recorded respectively. Videos of these flights are available in the video playlist[3] that belongs to this study. Tests without the extended detection rules result in many crashes and no successful flights. Especially in cases where an orange pole is present to the right of the vehicle, the chance of a detection failure turns out be too high to allow successful collision-free flights of several minutes.

4.6.3. Experiments in unadapted real word environments

Real flight tests were also performed in several unadapted rooms. Pictures of these rooms are shown in Figure 4.20. The first room is a meeting room with a long table with chairs in the middle. It contains three large white walls with little texture and

[3] https://www.youtube.com/playlist?list=PL_KSX9GOn2P987jTwx4szhPUPzpW5WJ3k

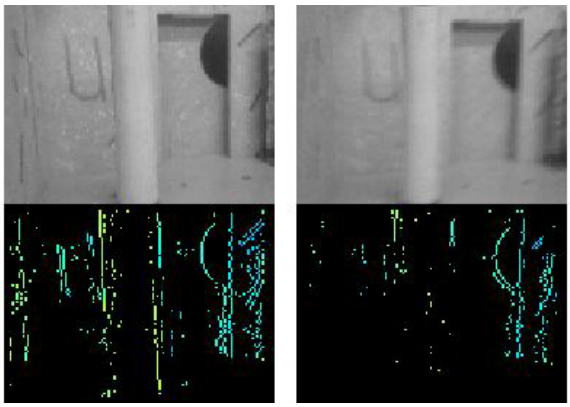

Figure 4.19: Camera images and corresponding disparity images of the same scene under different camera conditions. The examples show the difficulty of detecting the orange poles, which look like bright vertical beams (top images).
Left: camera is static. **Right:** camera is attached to the flying vehicle, thereby introducing motion blur.

one side with transparent windows. The second and third room are office spaces with screens surrounding the desks that form vertical obstacles in the middle of the rooms. The fourth space is a hallway with different features: curved walls, corners, corridor structures and several texture-poor walls.

In all tests, the DelFly was able to fly for as long as the battery permitted. Videos of these flights are available in the video playlist. Different flight behaviors were found in every room. In the meeting room the vehicle had the tendency to follow the contours of the room. In cases where the vehicle would get further away from the wall, the table or chairs were often interpreted as obstacles, making the vehicle to turn sooner and crossing the room. In this room the distance between the ceiling and the table, being less than two meters) is relatively small. Due to the variations in altitude caused by barometric pressure differences and small gusts due to a climate control system incorporated in the beams of the ceiling, this situation occurs very often. For this reason the operator manually corrected altitude errors to allow testing the robustness of the Droplet strategy in the horizontal plane. A flight time of 7 minutes was recorded in this room. In the two office spaces the amount of space between the desks is relatively narrow. This causes the vehicle to perform many turns, which results in a considerable reduction of flight time. Nevertheless, flight times of over 3 minutes were recorded in these rooms. It can be observed that the videos always stop when the battery gets empty. In this specific environment, this sometimes causes the vehicle to descend and hit an object with its tail. The hallway allows the vehicle to fly more straight parts, similarly to the meeting room, allowing also longer flight times. In this space, the vehicle virtually bounces between the walls. Because the vehicle always makes right turns, it stays within the space that is visible in the image. In some cases the vehicle flies into the corridor on the left, but it always turns around at some point when approaching the wall. Flight times of over 5 minutes have been recorded in this environment. It can

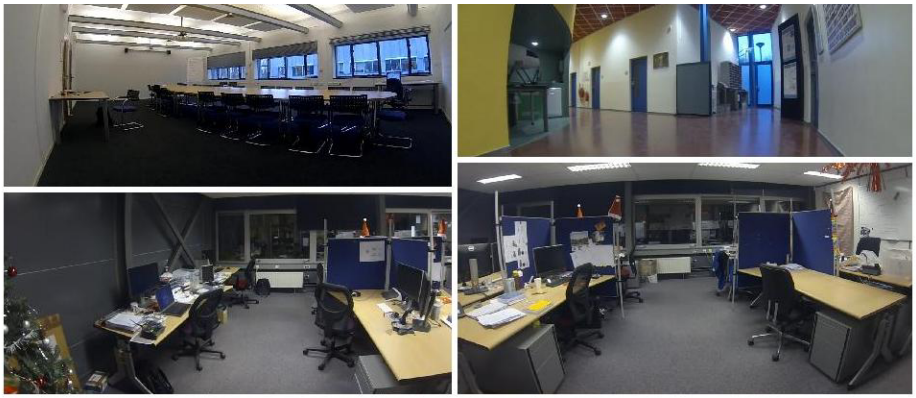

Figure 4.20: Images of three rooms where real flight tests were conducted.
Top-left: meeting room. **Top-right:** Hallway. **Bottom:** office spaces.

be observed that the vehicle may lightly touch a wall during an avoidance maneuver. This generally forms no problem for flapping wing MAVs. These situations can be prevented by applying trajectory following (instead of only timing the turn point) or by increasing the safety margin parameter of the Droplet area, but this makes the area larger and hence leads to reduced coverage.

4.7. Conclusions

In this study, a strategy is proposed for obstacle avoidance in small and cluttered environments, which takes into account both sensor limitations and nonholonomic constraints of flapping wing MAVs. The method relies on measured distances to obstacles from stereo vision. Its computational complexity, both in terms of time and space, makes it specifically suitable for use on small, embedded systems. Simulation experiments show that the method ensures collision avoidance in a small and cluttered room. Real flight experiments demonstrate that the method allows a 20 g flapping wing vehicle to autonomously perform sustained flight of up to 9 minutes while avoiding obstacles and walls in different environments.

References

[1] A. Bachrach, S. Prentice, R. He, and N. Roy, *RANGE–Robust autonomous navigation in GPS-denied environments,* Journal of Field Robotics **28**, 644 (2011).

[2] S. Grzonka, G. Grisetti, and W. Burgard, *A fully autonomous indoor quadrotor,* IEEE Transactions on Robotics **28**, 90 (2012).

[3] M. Nieuwenhuisen, D. Droeschel, M. Beul, and S. Behnke, *Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments,* Journal of Intelligent & Robotic Systems **1**, 1 (2015).

[4] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, *Visual odometry and mapping for autonomous flight using an RGB-D camera,* in *International Symposium on Robotics Research (ISRR)*, Vol. 2 (2011).

[5] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, *Vision-based autonomous mapping and exploration using a quadrotor MAV,* in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (IEEE, 2012) pp. 4557–4564.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, *ORB-SLAM: a versatile and accurate monocular SLAM system,* Robotics, IEEE Transactions on **31**, 1147 (2015).

[7] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-scale direct monocular SLAM,* in *European Conference on Computer Vision* (Springer, 2014) pp. 834–849.

[8] M. Pizzoli, C. Forster, and D. Scaramuzza, *REMODE: Probabilistic, monocular dense reconstruction in real time,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 2609–2616.

[9] C. Forster, M. Pizzoli, and D. Scaramuzza, *SVO: Fast semi-direct monocular visual odometry,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 15–22.

[10] R. J. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, *Autonomous MAV guidance with a lightweight omnidirectional vision sensor,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3856–3861.

[11] A. M. Hyslop and J. S. Humbert, *Autonomous navigation in three-dimensional urban environments using wide-field integration of optic flow,* Journal of guidance, control, and dynamics **33**, 147 (2010).

[12] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, *Learning monocular reactive UAV control in cluttered natural environments,* in *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (IEEE, 2013) pp. 1765–1772.

[13] A. Beyeler, J.-C. Zufferey, and D. Floreano, *Vision-based control of near-obstacle flight,* Autonomous robots **27**, 201 (2009).

[14] J.-C. Zufferey, A. Beyeler, and D. Floreano, *Optic flow to steer and avoid collisions in 3D,* in *Flying Insects and Robots* (Springer, 2010) pp. 73–86.

[15] J.-C. Zufferey, A. Beyeler, and D. Floreano, *Autonomous flight at low altitude with vision-based collision avoidance and GPS-based path following,* in *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (IEEE, 2010) pp. 3329–3334.

[16] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, *Implementation of wide-field integration of optic flow for autonomous quadrotor navigation,* Autonomous robots **27**, 189 (2009).

[17] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, *Combined optic-flow and stereo-based navigation of urban canyons for a UAV,* in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on* (IEEE, 2005) pp. 3309–3316.

[18] A. J. Barry and R. Tedrake, *Pushbroom stereo for high-speed navigation in cluttered environments,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 3046–3052.

[19] S. H. Lin, F. Y. Hsiao, C. L. Chen, and J. F. Shen, *Altitude control of flapping-wing MAV using vision-based navigation,* in *American Control Conference (ACC), 2010* (IEEE, 2010) pp. 21–26.

[20] F. Y. Hsiao, H. K. Hsu, C. L. Chen, L. J. Yang, and J. F. Shen, *Using stereo vision to acquire the flight information of flapping-wing MAVs,* Journal of Applied Science and Engineering **15**, 213 (2012).

[21] S. S. Baek and R. S. Fearing, *Flight forces and altitude regulation of 12 gram I-Bird,* in *IEEE RAS and EMBS Int Conf on Biomedical Robotics and Biomechatronics (BioRob)* (2010) pp. 454–460.

[22] R. C. Julian, C. J. Rose, H. Hu, and R. S. Fearing, *Cooperative control and modeling for narrow passage traversal with an ornithopter MAV and lightweight ground station,* in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2013) pp. 103–110.

[23] G. C. H. E. de Croon, K. M. E. de Clerq, R. Ruijsink, B. D. W. Remes, and C. de Wagter, *Design, aerodynamics, and vision-based control of the DelFly,* International Journal on Micro Air Vehicles **1**, 71 (2009).

[24] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, *Controlled Flight of a Biologically Inspired, Insect-Scale Robot.* Science **340**, 603 (2013).

[25] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov, *Development of the Nano Hummingbird: A Tailless flapping wing micro air vehicle.* in *50th AIAA Aerospace Science Meeting* (2012) pp. 6–12.

[26] E. F. Helbling, S. B. Fuller, and R. J. Wood, *Pitch and yaw control of a robotic insect using an onboard magnetometer,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 5516–5522.

[27] S. B. Fuller, E. F. Helbling, P. Chirarattananon, and R. J. Wood, *Using a MEMS gyroscope to stabilize the attitude of a fly-sized hovering robot,* in *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014* (Delft University of Technology, 2014).

[28] P. E. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, *Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor.* in *ICRA* (2012) pp. 4228–4235.

[29] S. S. Baek, F. G. Bermudez, and R. Fearing, *Flight control for target seeking by 13 gram ornithopter,* in *IEEE/RSJ Int Conf on Intelligent Robots and Systems.* (2011).

[30] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[31] L. Roberts, H. A. Bruck, and S. K. Gupta, *Autonomus loitering control for a flapping wing miniature aerial vehicle with independent wing control,* in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2014) pp. V05AT08A013–V05AT08A013.

[32] K. Bipin, V. Duggal, and K. Madhava Krishna, *Autonomous navigation of generic monocular quadcopter in natural environment,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 1063–1070.

[33] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, *Stereo vision-based obstacle avoidance for micro air vehicles using disparity space,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3242–3249.

[34] S. Xu, D. Honegger, M. Pollefeys, and L. Heng, *Real-time 3D navigation for autonomous vision-guided MAVs,* in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (IEEE, 2015) pp. 53–59.

[35] B. MacAllister, J. Butzke, A. Kushleyev, H. Pandey, and M. Likhachev, *Path planning for non-circular micro aerial vehicles in constrained environments,* in *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (IEEE, 2013) pp. 3933–3940.

[36] M. Pivtoraiko, D. Mellinger, and V. Kumar, *Incremental micro-UAV motion replanning for exploring unknown environments,* in *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (IEEE, 2013) pp. 2452–2458.

[37] A. A. Paranjape, K. C. Meier, X. Shi, S.-J. Chung, and S. Hutchinson, *Motion primitives and 3D path planning for fast flight through a forest,* The International Journal of Robotics Research **34**, 357 (2015).

[38] D. Dey, K. S. Shankar, S. Zeng, R. Mehta, M. T. Agcayazi, C. Eriksen, S. Daftry, M. Hebert, and J. A. Bagnell, *Vision and learning for deliberative monocular cluttered flight,* in *Field and Service Robotics* (Springer, 2016) pp. 391–409.

[39] S. Tijmons, G. C. H. E. de Croon, B. D. W. Remes, C. De Wagter, R. Ruijsink, E.-J. van Kampen, and Q. P. Chu, *Stereo vision based obstacle avoidance on flapping wing MAVs,* in *Advances in Aerospace Guidance, Navigation and Control* (Springer, 2013) pp. 463–482.

[40] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, *Visual-inertial navigation for a camera-equipped 25g nano-quadrotor,* Technical University, Munich (2015).

[41] D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,* International Journal of Computer Vision **47**, 7 (2002).

[42] X. Hu and P. Mordohai, *Evaluation of stereo confidence indoors and outdoors,* in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (IEEE, 2010) pp. 1466–1473.

[43] J.-C. Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano, *A 10-gram vision-based flying robot,* Advanced Robotics **21**, 1671 (2007).

[44] P. R. Cohen, *Empirical methods for artificial intelligence,* IEEE Intelligent Systems , 88 (1996).

5

Self-Supervised Learning applied to Autonomous Flight of lightweight MAVs

The previous chapters focused on using stereo vision for obstacle detection. Even though this approach is more reliable than optical flow, failures can still occur in special situations, such as when objects are transparent or small.

This chapter presents a method for distance estimation to objects by means of self-supervised learning. A monocular vision system is used which can also detect near-collisions. The camera image data that is recorded right before a near-collision, is used as training input for a learning algorithm that relates image appearance descriptors to estimated distances to the object. When the same object is approached later on, the learning algorithm can provide distance estimates which can be used for obstacle detection. The performance of this distance estimation method is tested in computer simulations, and applied to the task of obstacle avoidance. The findings from the simulations are validated in real test flights.

This chapter is based on the following article:
K. Lamers, **S. Tijmons**, C. De Wagter, G.C.H.E. de Croon, *Self-supervised monocular distance learning on a lightweight micro air vehicle*, In: International Conference on Intelligent Robots and Systems (IROS), (2016)

5.1. Introduction

Very small and lightweight Micro Air Vehicles (MAVs) can play an important role in many useful applications where size is important. Example applications are the inspection of difficult to reach areas, agriculture, monitoring, disaster management and tasks where interaction with humans is likely, since lightweight MAVs can be inherently safe. Most applications need fully autonomous systems, which requires onboard localization and navigation capabilities. Because sensor weight is a crucial factor for these small MAVs, the most promising solution involves the use of a single camera.

Many recent studies focus on odometry and mapping tasks using monocular approaches [1–3]. These methods provide accurate information about platform motion and structure of the environment, but are very demanding in terms of computing power. Furthermore these methods do not provide the scale of the estimated motions and distances.

For obtaining the scale from single images, learning techniques can be applied. Supervised learning has been applied by training based on a dataset that contains ground truth [4],[5]. But also self-supervised learning methods have been demonstrated that make use of terrain classification and specific system characteristics to combine information from different types of sensors [6–8]. Downsides of these approaches are either the use of heavy sensors for metric measurements, or assumptions such as ground plane visibility that are not generic for the application of MAVs.

This study proposes a self-supervised learning approach for monocular distance estimation that makes use of a very small short-range infrared sensor which serves as a near collision detector. By extracting efficient visual features from the camera image sequence before each near collision detection, the system learns the appearance of the object/environment at different distances through regression. This approach enables the 19 g flapping wing MAV from Fig. 5.1 to perform collision avoidance based on individual camera images without the need for additional continuous metric information and also enables it to adapt to its environment during flight.

The contribution of this study is: a self-supervised learning approach that relies on individual camera images and an efficient additional sensor for near collision detection. It is shown that this method can be implemented on a 2 g camera system to provide real time onboard distance estimates to a lightweight MAV.

Section 6.2 describes related studies. Section 5.3 explains the proposed self-supervised learning method. Section 5.4 discusses implementation details of the method and the setup of the experiments, which are performed in computer simulations (Section 5.5) and on the real platform (Section 5.6).

5.2. Related Work

Monocular vision is a commonly applied method for autonomous navigation of MAVs weighing less than 50 g. Since vision sensors are passive and provide a relatively high information density, various studies aim at relying solely on single cameras to

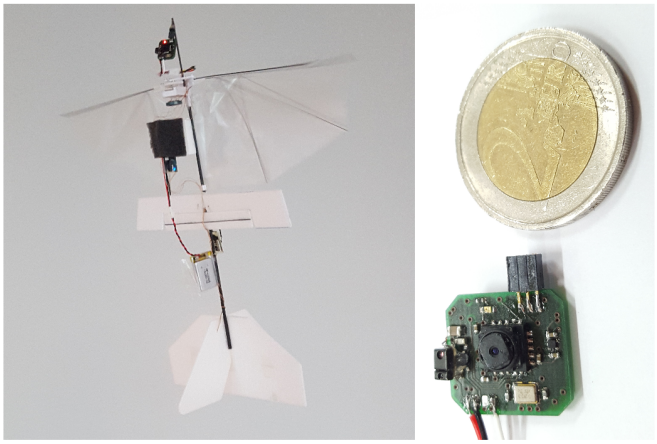

Figure 5.1: The DelFly platform as used in this study, based on the DelFly Explorer, featuring a monocular camera system attached on the nose. For more details on the DelFly Explorer, see Fig. 3.1. A closeup of the camera system is shown in the right part of the figure. The coin illustrates its small size. The camera board also contains a tiny infrared proximity sensor. This sensor serves as a near collision detector in a self-supervised learning scheme in which the DelFly learns to estimate distances based on obstacle appearance in the camera images.

perform various control and navigation tasks. Optical flow is an effective method that enables onboard processing on these small platforms, making them fully autonomous. So far, ego-motion estimation [9] and reactive obstacle avoidance [10][11] have been demonstrated using optical flow. In another study, a monocular Simultaneous Localization And Mapping (SLAM) method was demonstrated for hovering and waypoint navigation with such a small platform [12]. Processing was performed off-board in this study because of the high computational demand of this vision method. On extremely lightweight systems, even the processing of optical flow is still too demanding. Optical flow-based height control was demonstrated on a 101 mg platform by relying on off-board processing [13].

Monocular approaches can provide motion estimation and obstacle detection but these estimations lack direct scale measurements. The most common approach to compute scale is to use stereo vision. This is a computationally demanding method but has been demonstrated to run in real time on board several platforms [14],[15], even on the lightweight platform from Chapter 3. However, this approach requires a second camera, which increases the weight of the sensor payload significantly. The scale ambiguity can also be solved by relying on other secondary sensors. Several studies show monocular-SLAM approaches that use additionally an ultrasound and optical flow sensor facing downwards for measuring absolute horizontal speed [16]. A more elegant approach is to rely on the Inertial Measurement Unit (IMU), as this sensor is already present on many platforms for attitude control, which saves the weight of an additional sensor. By tightly-coupled fusion of IMU and monocular feature tracking measurements the scale problem can be solved [17]. Using this

approach is more difficult on a lightweight platform because of platform vibrations, and also when the platform has nonholonomic constraints. Another elegant solution is to purely use divergence estimates from monocular optical flow measurements by exploiting the self-induced oscillations that result from the fundamental imperfection of fixed-gain optical flow-based control [18]. This approach does not require additional sensors and has been demonstrated for vertical control of a quadrotor. It is theoretically plausible that also horizontal control can be integrated. The applicability to the platform in this study is less likely due to nonholonomic constraints and because the method requires fast platform dynamics.

Learning techniques have been applied in several vision based applications, also for MAV control. Imitation learning is a form of supervised learning that has been used to map monocular optical flow and visual features to control inputs given by a human pilot [19]. This method allows a quadrotor to avoid trees while flying in a forest. Another (nonlinear) supervised learning method has been used to perform stereo vision based distance estimation without the need to perform camera calibration [20]. Supervised learning has been used to solve the scale problem in monocular vision [4],[5]. Based on image databases with corresponding depth information (from laser or RGB-D measurements) this method learns how to select and use image features to obtain a dense depth map, but only for the trained environment. In self-supervised learning, the system generates its own reference data online. For example, this has been demonstrated on autonomous cars in two different ways. The first approach is to detect which part of the camera view corresponds to drivable road, based on short-range laser data [6]. After learning what the road looks like, the system determines from the camera images how the road continues at larger distances. The second approach is to assume that drivable road is visible right in front of the car and using the fact that the distance between the road and the mounted camera is fixed and known [8]. However, these assumptions cannot be used on flying vehicles. Self-supervised learning has also been demonstrated for the landing task of an MAV [21]. Optical flow information is obtained to detect surface discontinuities while the MAV is moving around. Objects and potential landing locations are then classified and their appearance is learned. When the drone has to land, it can choose landing locations from still images. Combining optical flow and appearance has also been shown in an application where a wheeled robot learns from near collisions with trees [7]. When its infrared sensors detect a near tree, optical flow from a history of images is then used to track it over time and to learn the appearance of the tree for a range of distances. Experiments show that the average number of tree encounters and the time to travel a certain path both decrease significantly using this learning approach. Again the assumption of a ground plane is used to estimate distances.

5.3. Self-Supervised Learning for Distance Estimation

In this study, a Self-Supervised Learning (SSL) method is proposed that learns to estimate distances from still images. SSL differs from classical supervised learning

in that labels are not generated by a human but by the robot itself. Essentially, SSL allows different sensors to work together: measurements on a certain parameter obtained from one sensor are used to label data from a second sensor which does not directly have knowledge about this parameter. When enough training data is collected, this method enables the robot to use only the second sensor to measure this parameter using regression. In this study self-supervised learning is applied to combine a short-range sensor that provides binary distance information with a camera that provides continuous data at all distances. This combination of sensors will provide distance estimates also for longer ranges.

5.3.1. Distance estimation methods

Distance measurements are performed in two ways: using camera images in combination with information from the learning process or using a proximity sensor for detecting near collisions. The first method, using the camera, requires that the system can rely on training data that was learned in the past. The second method, using the proximity sensor, only indicates whether the distance to a nearby obstacle is too small to continue flying in the current direction. In this case, two simultaneous actions are performed: the vehicle changes its heading, and recently recorded images are used to perform an iteration in the learning process.

Camera

A TCM8230 color camera provides RGB images throughout each flight. To enable distance estimation on the limited processor (168 MHz, 192 kB), data reduction of the images is realized using an efficient image descriptor which has the form of a histogram. The histogram indicates the frequencies of a predefined set of textons [22]. Textons are fundamental micro-structures in images. In this study, a set of R small representative RGB image patches form a dictionary of textons. From each camera image, N evenly spread patches with the same size as the textons are extracted and matched with the texton dictionary, based on minimum Euclidean distance. The indices of all best matches with the dictionary form the histogram of texton occurrences for each image. Note that the histograms contain information about the overall appearance of images, not on local image patterns. The histograms are used for two purposes. First, to obtain a distance estimate for the current image based on what has been learned (Section 5.3.2 explains how this is done). Second, to serve as temporary training data in case a near collision occurs shortly after the image was recorded.

Proximity sensor

A TMG399x infrared proximity sensor (2×4 mm) is used as a short-range binary detector for near collisions; it indicates whether an object is detected within a range of approximately 50 cm which allows the MAV to perform an evasive maneuver. In case a near collision is detected, the maneuver is executed and recently stored histograms are assigned a distance label. The distance assigned to each histogram is based on retrograde extrapolation assuming constant heading, constant flight speed and constant frame rate. These distances are regarded as ground truth and are used to perform an iteration in the learning process.

5.3.2. Learning Algorithms

The effectiveness of different learning algorithms is tested in this study. These algorithms have two functions. First, to provide a distance estimate based on a histogram input. Second, to learn from near collisions by importing histograms with assigned distance labels as training data.

Perceptron Network

The simplest approach that was tested is an ADALINE network, which is a single-layer perceptron without hard limits. A perceptron is a simple form of a neural network in which the output **a** (distance estimate) is the weighted sum of all inputs **p** (the $R \times 1$ histogram) and a bias term:

$$\mathbf{a} = \mathbf{W}\mathbf{p} + \mathbf{b} \tag{5.1}$$

When a near collision occurs, the weights **W** $(1 \times R)$ and bias **b** are updated with a Widrow-Hoff learning rule [23]:

$$\begin{aligned} \mathbf{W}(k+1) &= \mathbf{W}(k) + 2\alpha\mathbf{e}(k)\mathbf{p}^{\mathbf{T}}(k) \\ \mathbf{b}(k+1) &= \mathbf{b}(k) + 2\alpha\mathbf{e}(k) \end{aligned} \tag{5.2}$$

In this equation, α is the learn rate and **e** is the error between **a** and the corresponding ground truth label.

k-NN

k-Nearest Neighbours (k-NN) is an algorithm that can be used for both classification and regression problems [24]. In k-NN regression an input feature vector is compared with the full set of trained feature vectors and the k nearest neighbours (based on smallest Euclidean distances) are used to calculate the output using the labels of the training samples. In this study, the feature vectors are formed by the image histograms that are labeled with distance values. In case k>1, distances are estimated by taking the average of the corresponding distance labels. k-NN is a type of lazy learning; new training data (histograms with distance labels) is simply added to the training set. This makes the training phase fast, but leads to large amounts of training data that needs to be stored and a slow (distance) evaluation process.

k-NN with clustering

To solve the mentioned limitations of the k-NN algorithm, a clustering method is proposed that reduces the amount of stored training data. Similar methods, such as condensation [25] or instance selection [26], have been proposed to remove either noisy samples from the training set to improve accuracy, or to eliminate redundant samples to optimally reduce the size of the training set. The currently proposed clustering method is based on the assumption that similar histograms correspond to similar parts of an environment, and that merging their feature vectors and labels is therefore legitimate. This allows for storage of a fixed amount of training data while maintaining diversity. Clustering is done by looking for pairs of similar histograms (based on Euclidean distance) and by only storing the averages of their histogram values and labeled distances.

Figure 5.2: **Top**: screenshot showing the simulated environment. **Bottom**: photo of the DelFly flying in the test room. The simulated room is an imitation of the real test room.

5.4. Implementation and Test Setup

The proposed distance estimation method is used to enable obstacle avoidance on a lightweight flapping wing MAV, the DelFly [27, 28]. In this study the vehicle has a wing span of 28 cm and a weight of 17 g. This includes a 1 g autopilot with an IMU (MPU9150) and a barometer. Its payload is a 2 g camera system featuring a TCM8230 color camera, a TMG399x infrared proximity sensor and an STM32F405 ARM processor. The system has a total weight of 19 g and is able to run the learning algorithm on board.

Its flight characteristics make the DelFly a suitable platform to use the proposed distance estimation method because it flies passively stable with a constant low forward speed. The vertical speed and distance to the ground can be restricted using feedback from the barometer. At low speed the vehicle can perform avoidance maneuvers within the space covered by the short-range proximity sensor, such that no real crashes occur. Using gyroscope feedback a fixed heading can be maintained when no control action is required.

The camera system is mounted in a specific way to the DelFly; the proximity sensor is aligned with the forward velocity vector and thus looks straight ahead. The camera is rotated with an offset of 15 deg to the left. The following sections on simulation and flight tests explain how this setup is exploited for control purposes. Tests haven been performed in a small room of 4×4 m which has walls with different types of textures, as shown in Fig. 5.2. The top image in this figure shows the test room as used for simulations. For the 6×6 m simulated room photos from the real room are used.

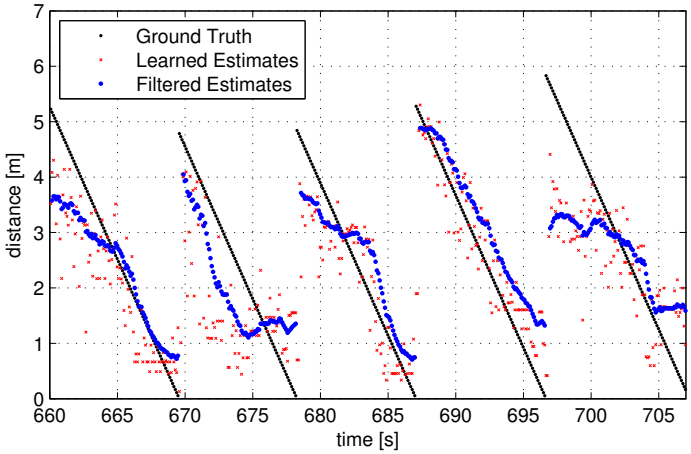

Figure 5.3: Estimated distance with k-NN using 500 clustered points after 660 s of training.

5.5. Simulations

Computer simulations were initially performed to test and analyse the performance of the different learning-algorithms and to explore effective reactive avoidance strategies.

5.5.1. Distance estimation performance

To compare the different learning algorithms, the vehicle is simulated as flying at a constant height, in straight lines and with a speed of 0.55 m/s. Each time the vehicle hits a wall, its heading is changed instantly with a random offset such that it continues flying within the test room. This way a data set with recorded image histograms (10 Hz) and flight tracks is obtained to test the learning algorithms.

Fig. 5.3 shows an example of the distance estimation performance of k-NN learning with clustering after 660 s of training.Individual distance estimates are presented, as well as the result after low-pass filtering. It is observed that the filtered estimates show an obvious correspondence with the ground truth data, even though the estimates contain significant noise. For this reason the performance of the learning algorithms is expressed by the correlation coefficient, which is a measure of the linear dependence between the ground truth data points and the estimated data points.

Fig. 5.4 shows the correlation coefficients for the different algorithms over time. The most interesting observation is that the clustered k-NN method outperforms the Widrow-Hoff method and has almost similar performance as the standard k-NN method. The correlation coefficient increases significantly during the first few minutes, and reaches a more steady performance afterwards. The effectiveness of the clustering method is clearly visible from Fig. 5.4. For comparison the performance is shown if a fixed number of histograms would be maintained by simply dumping

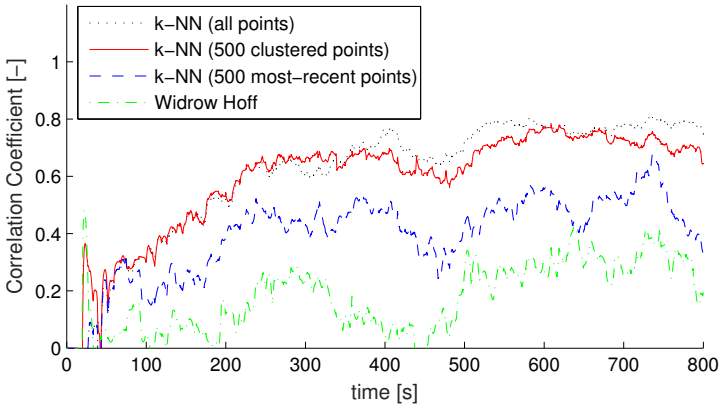

Figure 5.4: Correlation between distance estimate and ground truth using different algorithms. For k-NN with all points, all histograms are stored which after 600 seconds adds up to 6000 data points. Both other k-NN methods keep only 500 data points in memory. The clustered approach is also compared to an approach that simply remembers the most recent points. All k-NN methods use k=5. For the Widrow-Hoff approach, α=0.05. All methods use histograms with R=24 bins constructed from matching N=70 patches. The patch size is 5×5 pixels.

the oldest histograms. After reaching the maximum number of histograms (50 s) the difference in performance becomes visible. The effectiveness of the clustering method can be explained by analysing the histogram data using the t-SNE [29] algorithm. It allows to visualize the total set of high dimensional histograms as a two-dimensional image, as shown in Fig. 5.5. Each histogram is a point in this image, and relative distances between the points are based on similarity. The result shows that clusters of similar points are formed, and that within the groups color gradients are visible. This confirms the hypothesis that similar histograms correspond to the same part of the test room and that the histograms change gradually with distance. The proposed clustering method is thereby justified and used in further experiments.

5.5.2. Control

The camera is mounted with an offset on the vehicle, also in simulation. This allows for obtaining individual distance estimates (k-NN with 500 clustered points) for the two halves of the camera images: one halve that looks straight ahead and one halve that looks to the left side. Control is based on thresholds on the two distance estimates; if either of the two estimates indicates a small distance, the vehicle turns left with constant input. Otherwise the vehicle flies straight. The threshold used for the left side is more conservative to ensure a free space on the side to perform an avoidance maneuver.

Fig. 5.6 shows results of two tests where the vehicle uses its distance estimates for avoidance control. In the first test, avoidance control is active from the start, in the second test the switch to active avoidance is made after 300 s. In both tests

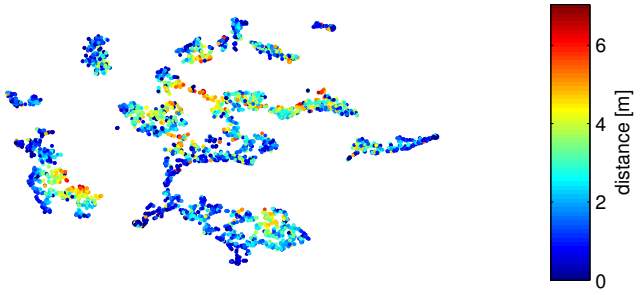

Figure 5.5: t-SNE: 3000 histograms with 30 texton bins. The histograms are represented as colored dots. The colors indicate corresponding distances. The 2-D locations of the dots do not have an absolute meaning, but their relative distances indicate their similarity.

training is performed throughout the flight. For the second test, only the flight trajectory after the control switch is shown. From the results it is clear that in the second test, the near collision rate after 300 s is much lower than in the first test. This can be explained by the total number of near collision events which is much higher at this point. Furthermore, the flight track of the first test indicates that the vehicle flies a lot of small circles which slows down the learning process. The moment of switching apparently influences the total number of near collision events that occurs within a certain amount of flight time.

5.6. Experimental Results

The k-NN learning method using 500 clustered points has been implemented on the 2 g monocular vision system of the DelFly. First tests show the potential performance of this system by manually walking around in the test room. In these tests the operator walks in straight lines, and chooses a new random direction when the proximity sensor indicates a near collision. Results of these performance tests are shown in Fig. 5.7. The correlation coefficient reached after 600 s is comparable to what was observed in the simulation results. Fig. 5.8 shows distance estimates during one of the tests after a correlation coefficient of more than 0.5 was reached. These results clearly visualize that wall approaches can be detected.

Autonomous flight tests with the DelFly have been performed using the proposed learning approach. In these tests, the altitude is regulated using barometer feedback and the heading is controlled using gyroscope feedback (for enabling straight flight paths during wall approaches). Furthermore, a visual tracking system is used for logging the position of the vehicle and to assist in deciding to turn left or right in case of a near collision detection. Especially in case the wall is approached non-perpendicular, it is critical to turn in the right direction. This assistance can be made superfluous by increasing the heading control authority of the vehicle at low speeds. Fig. 5.9 shows distance estimation performance results of three different flights. In these tests the trained data in the camera system is cleared prior to the

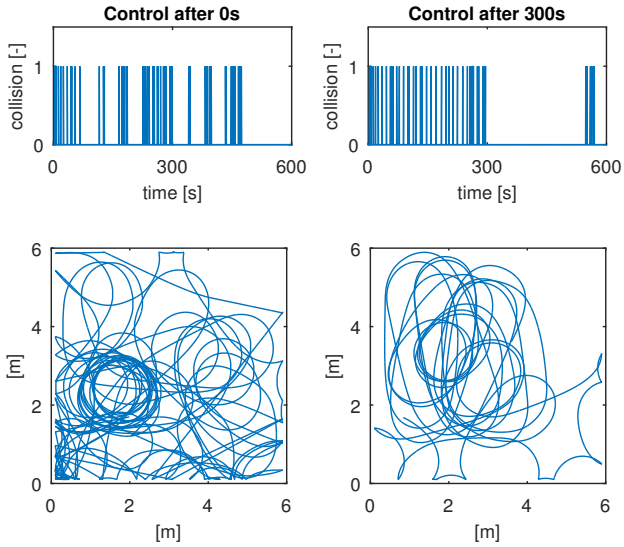

Figure 5.6: Near collision events and flight trajectories of two simulated flights where vision-based control is applied. The left plot shows results when control is active from the start. The right plot shows results when control is activated after 300 seconds. The bottom plots show trajectories from the moment vision-based control was turned on.

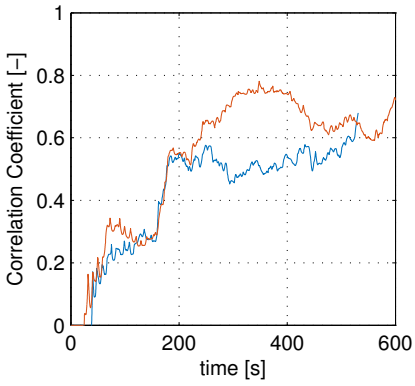

Figure 5.7: Correlation coefficient of learned distance estimates over time for two different runs while walking with the DelFly through the test room.

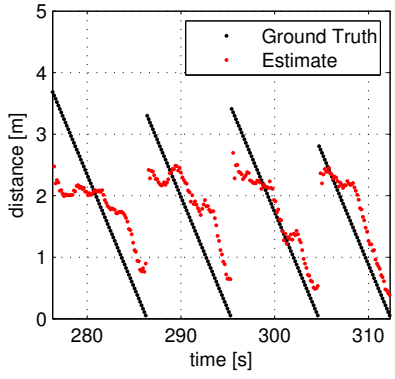

Figure 5.8: Partial results showing estimated distance versus ground truth data from a test while walking with the DelFly.

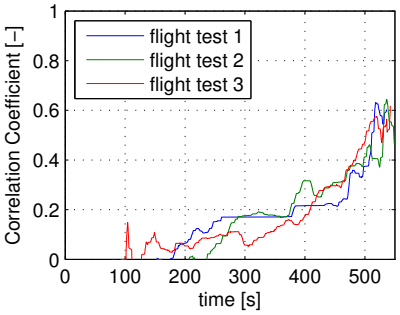

Figure 5.9: Correlation coefficient of learned distance estimates over time for three flight tests with the DelFly using closed loop heading control.

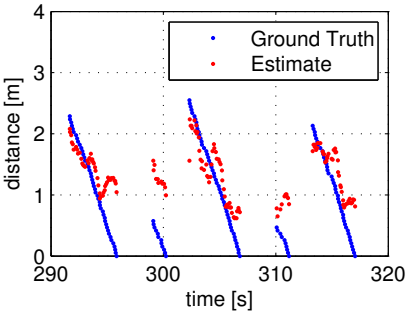

Figure 5.10: Partial results showing estimated distance versus ground truth data from a real flight test with the DelFly using closed loop heading control.

flight. Note that lower correlation coefficients are reached as in previous experiments (0.5-0.6 instead of 0.6-0.7), and that the learning rate is lower. Fig. 5.10 shows distance estimates during the final part of one of these test flights. These results also show a worse performance compared with previous tests, especially for the small ranges. This can be explained by variations in the altitude, heading changes due to disturbances, and platform vibrations. These influences result in variations in camera observations, especially at distances close to the walls.

To show the potential of the proposed learning method the autonomous tests were split in two parts. First a training flight was performed to train the camera system in the test room. In this test, the vehicle only performs collision avoidance maneuvers based on inputs from the proximity sensor. A flight trajectory of such a test is shown in Fig. 5.11. After the training flight the trained data is retained for the next flight. During this subsequent flight the distances estimated by the learning algorithm are used as control input. A threshold value of 1.2 m is used to decide whether to fly straight or to turn left. The vehicle always turns left, because the camera is slightly rotated in this direction and never observes obstacles to the right of the vehicle. The results are shown in Fig. 5.12.

5.7. Conclusions

A self-supervised learning method is proposed that enables a lightweight MAV to estimate distances based on monocular images. The method combines distance information from a small infrared proximity sensor during near collisions with an efficient image description algorithm to enable online distance estimation on a 2 g camera system. The k-NN based learning method uses a clustering step to limit the amount of stored training data that has a marginal effect on performance. Computer simulations show that the proposed method allows the MAV in this study to significantly reduce the number of near collisions over time. Real world tests indicate that similar performance can be reached on the real system. Real test flights indicate a lower learning rate, but show that collision avoidance is possible

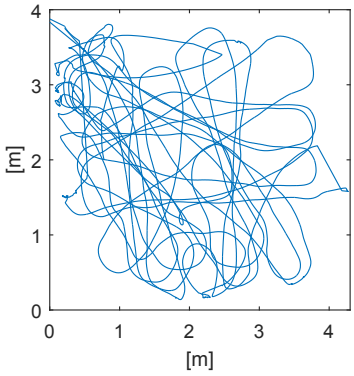

Figure 5.11: Flight trajectory of training test flight of the DelFly equipped with the monocular camera system.

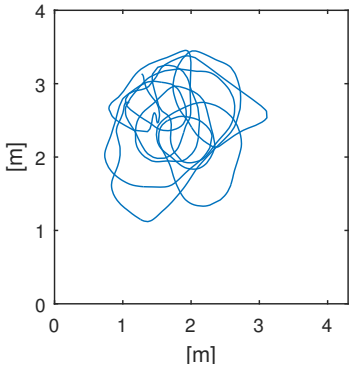

Figure 5.12: Flight trajectory of the DelFly with monocular distance estimates in the loop for active wall avoidance.

using the proposed method.

References

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, *SVO: Fast semi-direct monocular visual odometry,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 15–22.

[2] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-scale direct monocular SLAM,* in *European Conference on Computer Vision* (Springer, 2014) pp. 834–849.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, *ORB-SLAM: a versatile and accurate monocular SLAM system,* Robotics, IEEE Transactions on **31**, 1147 (2015).

[4] C. Plagemann, F. Endres, J. Hess, C. Stachniss, and W. Burgard, *Monocular range sensing: A non-parametric learning approach,* in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* (IEEE, 2008) pp. 929–934.

[5] K. Bipin, V. Duggal, and K. Madhava Krishna, *Autonomous navigation of generic monocular quadcopter in natural environment,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 1063–1070.

[6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, *Self-supervised monocular road detection in desert terrain.* in *Robotics: science and systems* (Philadelphia, 2006).

[7] A. Lookingbill, D. Lieb, and S. Thrun, *Optical flow approaches for self-supervised learning in autonomous mobile robot navigation,* in *Autonomous Navigation in Dynamic Environments* (Springer, 2007) pp. 29–44.

[8] B. Lee, K. Daniilidis, and D. D. Lee, *Online self-supervised monocular visual odometry for ground vehicles,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 5232–5238.

[9] A. Briod, J.-C. Zufferey, and D. Floreano, *Optic-flow based control of a 46g quadrotor,* in *Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments, IROS 2013*, EPFL-CONF-189879 (2013).

[10] R. J. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, *Autonomous MAV guidance with a lightweight omnidirectional vision sensor,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3856–3861.

[11] J.-C. Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano, *A 10-gram vision-based flying robot,* Advanced Robotics **21**, 1671 (2007).

[12] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, *Visual-inertial navigation for a camera-equipped 25g nano-quadrotor,* in *IROS2014 aerial open source robotics workshop* (2014) p. 2.

[13] P.-E. J. Duhamel, N. O. Pérez-Arancibia, G. L. Barrows, and R. J. Wood, *Altitude feedback control of a flapping-wing microrobot using an on-board biologically inspired optical flow sensor,* in *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (IEEE, 2012) pp. 4228–4235.

[14] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, *Stereo vision-based obstacle avoidance for micro air vehicles using disparity space,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 3242–3249.

[15] A. J. Barry and R. Tedrake, *Pushbroom stereo for high-speed navigation in cluttered environments,* in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015) pp. 3046–3052.

[16] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers, *Collision avoidance for quadrotors with a monocular camera,* in *Experimental Robotics* (Springer, 2016) pp. 195–209.

[17] S. Shen, N. Michael, and V. Kumar, *Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 5303–5310.

[18] G. C. H. E. de Croon, *Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy,* Bioinspiration & biomimetics **11**, 016004 (2016).

[19] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, *Learning monocular reactive UAV control in cluttered natural environments,* in *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (IEEE, 2013) pp. 1765–1772.

[20] F. Sinz, G. Quinonero-Candela, G. Bakir, C. Rassmussen, and M. Franz, *Learning depth from stereo,* in *Pattern Recognition Proc 26th DAGM Symposium LNCS 3175* (2004) pp. 245–252.

[21] H. W. Ho, C. De Wagter, B. D. W. Remes, and G. C. H. E. de Croon, *Optical-flow based self-supervised learning of obstacle appearance applied to mav landing,* arXiv preprint arXiv:1509.01423 (2015).

[22] M. Varma and A. Zisserman, *Texture classification: Are filter banks necessary?* in *Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on*, Vol. 2 (IEEE, 2003) pp. II–691.

[23] B. Widrow, M. E. Hoff, *et al.*, *Adaptive switching circuits,* in *IRE WESCON convention record*, Vol. 4 (New York, 1960) pp. 96–104.

[24] N. S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression,* The American Statistician **46**, 175 (1992).

[25] F. Angiulli, *Fast condensed nearest neighbor rule,* in *Proceedings of the 22nd international conference on Machine learning* (ACM, 2005) pp. 25–32.

[26] Á. Arnaiz-González, M. Blachnik, M. Kordos, and C. García-Osorio, *Fusion of instance selection methods in regression tasks,* Information Fusion **30**, 69 (2016).

[27] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[28] C. De Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. de Croon, *Autonomous flight of a 20-gram flapping wing MAV with a 4-gram onboard stereo vision system,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 4982–4987.

[29] L. v. d. Maaten and G. Hinton, *Visualizing data using t-sne,* Journal of Machine Learning Research **9**, 2579 (2008).

6

A Flight Control Mechanism for Flapping Wing MAVs that allows Agile Maneuvers

The previous chapters presented effective methods for obstacle avoidance on the DelFly flapping wing MAV. However, there will always remain specific situations in which obstacles will not be detected sufficiently in advance. In those situations, the vehicle should have the capability to reverse its heading quickly on the spot to prevent a collision.

In this chapter, a control mechanism design is proposed that allows the vehicle to perform a fast heading-reversal while hovering. It is experimentally analyzed what control forces and moments can be produced by this design, and how it can be implemented on a real flapping wing MAV (the DelFly II). Results from test flights are presented that show the ability of the vehicle to perform the fast heading-reversal maneuvers. Furthermore, an implementation of this system is presented that enables and demonstrates tailless flight.

This chapter is based on the following articles:
S. Tijmons, M. Karásek, G.C.H.E. de Croon, *Attitude control system for a lightweight flapping wing MAV*, In: Bioinspiration & Biomimetics (SUBMITTED)

J.L Verboom, **S. Tijmons**, C. De Wagter, B.D.W. Remes, R. Babuska, G.C.H.E. de Croon, *Attitude and altitude estimation and control on board a Flapping Wing Micro Air Vehicle*, In: IEEE International Conference on Robotics and Automation (ICRA), (2015)

6.1. Introduction

As the application areas for Micro Air Vehicles (MAVs) are expanding, more concepts for platform designs are being explored. While fixed-wing concepts were the dominant platform for decades, multirotor concepts have become enormously popular over the last few years. A major trend that can be observed is that these concepts are combined in hybrid designs [1–4]. This enables MAVs to take off and land vertically while also being able to perform long endurance forward flight.

Flapping wing MAV research is focused on enabling the same combination of vertical and horizontal flight by drawing inspiration from nature. While hybrid concepts use different types of devices to produce thrust forces (rotors), lift forces (wings) and control moments (rotors and control surfaces) flapping wing MAVs can generate both forces and control moments by their wings only. This approach is inspired by the flight methods of various birds and insects that show a wide range of capabilities: extremely precise hovering, agile manoeuvring, efficient forward flying and gliding, and fast transitioning between flying, hovering and landing.

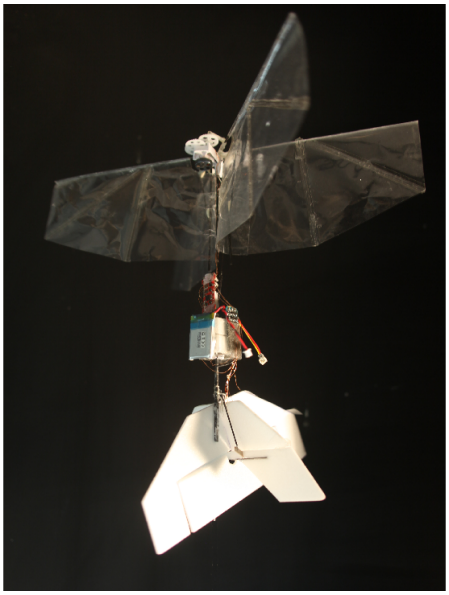

Figure 6.1: DelFly II flapping wing MAV with wing vectoring control system as developed and analysed in this study. The image shows the vehicle pose that corresponds to hovering flight.

Various flapping wing MAV designs have been demonstrated with varying flight capabilities. Some studies focus only on forward flight, mainly using relatively heavy bird-like systems that have a weight of a few hundred grams [5, 6]. In the same weight class an imitation of a dragonfly has been demonstrated that can perform both hovering flight and forward flight [7]. On smaller systems, that have a weight of a few tens of grams, the main research focus is on performing stabilized hovering and steering control, which is especially challenging for designs without a tail [8–

12] because of their intrinsic instability. In these studies, different methods are proposed for using the wings as control devices. Other systems use a tail [13–17], which provides (partial) inherent stability. In these approaches, control surfaces are added to the tail to allow active stabilization and to enable steering commands.

While there is a clear stability advantage when using a tail, the use of control surfaces has an important drawback. The DelFly II, shown in Fig. 6.1, is a lightweight flapping wing MAV that originally uses a tail with rudder and elevator [18], and the version from Chapter 3 also uses aileron surfaces close behind the wings for improved heading control. Previous research on this platform has shown that at very low speeds and especially during hovering, the effectiveness of the control surfaces diminishes drastically [19]. It was observed that this limits the turn rate significantly and results in poor attitude control performance in case of disturbances.

In this study a wing control mechanism is proposed that is added to the flapping mechanism of the DelFly II MAV [20]. It allows for independent vectoring of the wing pairs on both sides. The mechanism can also control the tension in the two wing pairs. The proposed wing control mechanism primarily replaces the function of the ailerons and elevator to allow controlled hovering flight. The mechanism allows the DelFly to fly at low velocities, which is very useful for autonomous flight. For instance, situations can occur where the vehicle needs to turn around immediately to prevent a collision. This requires the ability to hover and turn on the spot. It is shown in the current study that the proposed wing control mechanism improves the hovering performance and allows for fast turns in this flight condition. In addition to these capabilities, this study also shows that the proposed wing control mechanism can be used to control all three attitude angles required for hovering. The mechanism provides sufficient control authority to enable tailless hovering.

6.2. Related Work

This section discusses studies in the field of flapping wing MAVs that relate to control of such vehicles. First, different studies on attitude control mechanisms are discussed. Second, work on autonomous attitude control on board of such vehicles is discussed. Third, the few studies concerning autonomous flight tasks using onboard systems are discussed.

6.2.1. Wing control techniques for attitude control

A straightforward method for control of flapping wing MAVs is to use a tail with control surfaces [13–16], similar to fixed-wing aircraft. The wings purely generate thrust and lift in such setups. Extra aileron surfaces can be applied to obtain improved yaw torques for heading control as explained in Sec. 3.2. By using a tail (or another form of damping surface), the vehicle obtains passive stability in forward flight. However, the stabilizing effect of additional surfaces decreases significantly at low speeds. Furthermore, a downside of adding these surfaces is that the vehicle becomes more susceptible to wind disturbances.

Flapping wing MAV research focuses on flight control methods that are inspired

by nature. Most insects mainly rely on their wings for flight control, especially in (near) hover conditions. Wings can be used for stabilization and steering by deforming their shape and by modulating their angle of attack and their kinematics. Different combinations of these techniques have been implemented and tested on various flapping wing MAVs.

It was shown experimentally that controlling the feathering angle and camber of a wing by pulling at a point on the wing can produce a pitching moment [21]. This effect was only demonstrated on a static setup. Fully tailless control was demonstrated on the $19\ g$ Nano Hummingbird that uses a combination of wing twist modulation and rotation modulation [8]. In this system, wing twist is controlled by pulling the wing root away from or towards the wing membrane. This regulates the tension in the wing, and indirectly twists its camber, allowing the vehicle to roll. Rotation modulation is obtained by tilting the wing root forward and backward, which affects the angle of attack of the wing. This can be done either symmetrically (for pitch) or differentially (for yaw). Similar wing twist modulation and rotation modulation effects are obtained using a different mechanism on the KUBeetle [12]. This tailless platform of $21\ g$ demonstrates 14 seconds of stable hovering flight.

Control by wing kinematics modulation has been shown in real (tethered) test flights on the extremely small $0.1\ g$ Robobee [9]. The wings are individually actuated by piezoelectric actuators. By changing the wing stroke amplitudes, roll control is obtained. Moving the mean stroke angle of the wings forward or backward results in pitching torques. Yaw control is implemented by modulating the stroke velocity within a wing beat. Similar control methods were tested on a larger system driven by a single DC motor [10]. Using a mechanical solution in the flapping mechanism this system can control the wing stroke amplitude (for roll) and mean stroke offsets (for pitch). Tests were performed in a static setup, showing that the system provides adequate torques. Another study shows real test flights with a Robotic Hummingbird platform that uses wing kinematics modulation [11]. As in the previous two studies, roll control is obtained by changing the stroke amplitude of the left and right wing. Furthermore, a hinge tilting mechanism is used to control the angle of the leading edge stroke planes of the left and right wing individually, which allows for pitch and yaw control. This design weighs $62\ g$, which is relatively heavy compared to the other systems that were discussed so far, which have a weight of $20\ g$ or less. A different mechanism for changing the leading edge stroke plane is presented which makes use of rotating leading edge flapping slots [22]. This allows for generation of pitch and yaw commands. In combination with a trailing edge shift mechanism, roll control is also obtained. This mechanism is tested on a force balance, showing that all three control moments can be generated.

Two heavier systems, both weighing more than $100\ g$, obtain control via two other systems. The BionicOpter ($175\ g$) uses four wings that are driven by a central motor [7]. For each wing the stroke amplitude and the stroke plane tilt angle can be controlled. This system is capable of hovering and transitioning to forward flight. The Robo Raven is a bird-like platform of $312\ g$ [6]. It uses two servos to drive its two wings and an extra servo to actuate the tail as a ruddervator. This platform performs forward and gliding flight. Furthermore, a new design concept, the quad-

thopter [23], has been introduced which consists of four independently controlled flapping mechanism. By flapping these mechanisms at individual frequencies, attitude control can be performed in a similar way as for normal quadrotors. Flight times of 9 minutes and more have been realized with this $33\ g$ tailless platform.

6.2.2. Autonomous attitude control

Autonomous attitude control was realized on the DelFly II in a previous study [19]. In that study, the vehicle design includes a tail with active rudder and elevator surfaces. Aileron surfaces are used in this design for heading control during hover. This study demonstrates the low effectiveness of the control surfaces during hover.

On tailless platforms, several studies focus on realizing attitude stabilization and control. Attitude rate stabilization was shown on the Nano Hummingbird [8] using gyroscopes. Attitude control was still performed manually. Similarly, gyroscope feedback stabilization has been demonstrated on the Robobee [24], allowing the vehicle to keep its hover attitude for several seconds. On both platforms attitude estimation using accelerometers was not possible due to platform vibrations. In another study pitch and yaw control was realized on the Robobee using magnetometer feedback [25]. Full attitude stabilization and control was demonstrated in test flights with the $62\ g$ Robotic Hummingbird [11]. In this study, attitude estimation was realized using feedback from accelerometers. The tailless $21\ g$ KUBeetle [12] demonstrated autonomous attitude control by pitch,roll and yaw rate control based on feedback from gyroscopes. The tailless quad-thopter platform [23] is able to perform full attitude control autonomously by pitch, roll and yaw angle control based on gyroscope and accelerometer feedback.

6.2.3. Autonomous flight capabilities

Some of the discussed studies show autonomous control capabilities of flapping wing MAVs. Full autonomous flight has been shown in Sec. 5.6 with the Delfly Explorer, which in terms of actuation is similar to the DelFly II. This is a system with a tail and aileron surfaces. It uses gyroscope feedback for stability enhancement of pitch and yaw and a barometer for height control. Furthermore it has a $4\ g$ stereo camera system to perform onboard obstacle avoidance, enabling sustained autonomous flight in unknown environments for up to 9 minutes. Target-seeking using an onboard Wii-mote infrared camera has been demonstrated with a tailed platform [26]. In that study, pitch and yaw angle control are performed autonomously. The bird-like Robo Raven platform [6] demonstrated GPS/IMU-based loitering. This entails heading control to steer the vehicle to a specified GPS-waypoint.

6.3. Control Mechanism

A new mechanism for wing actuation is presented that creates control moments for attitude stabilization and control. This mechanism is specifically suitable for four-winged flapping wing MAVs such as the DelFly. The mechanism is combined with the original flapping mechanism of the DelFly, which has proven to be a reliable and efficient design. The kinematics of the wing leading edges are therefore the same

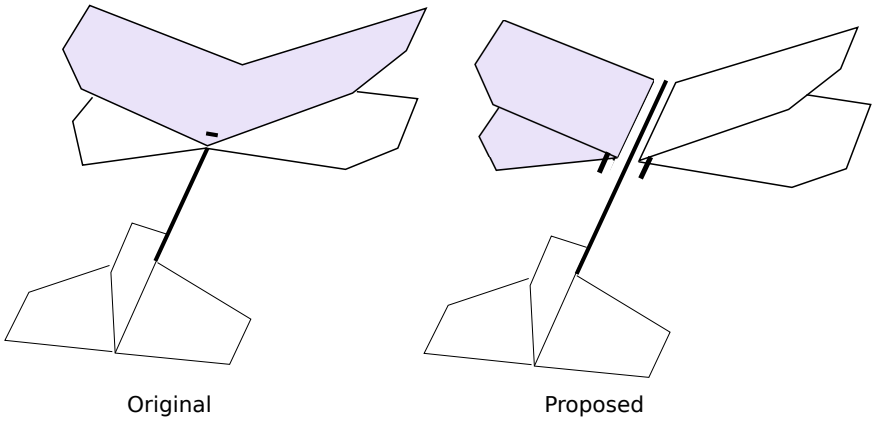

Figure 6.2: Comparison of wing attachment for original DelFly II design (left) and the design as proposed in this study (right). The left image shows how the top and bottom wings for the original DelFly II are attached to the fuselage using a hook on the trailing edge. The right image shows how the left and right wing are folded around the left and right wing roots, respectively. In this implementation the top left and bottom left wing parts are connected and form a single left wing. The same goes for the right wing.

as in the original DelFly II design. However, the wings are attached in a different way. This is illustrated in Fig. 6.2. In the original design, a bottom and a top wing are used that are fixed to the fuselage at the centers of their trailing edges. In the proposed design, use is made of a left wing and a right wing. These wings are not fixed, but are folded around carbon rods that act as wing roots.

Control moments are generated by actuating the two wing roots individually. Fig. 6.3 indicates the degrees of freedom (red arrows) of the two roots (black arrows). Both wing roots have two degrees of freedom around the same axis (dashed line): a rotation and a translation. By combining these degrees of freedoms all three attitude control moments can be generated, which is shown in Fig. 6.4.

This method of actuating the wing roots shows similarities with the systems of the Nano Hummingbird [8] and the KUBeetle [12]. This is especially true for roll, as the same principle is used: by moving the wing roots inward and outward the tension in the wing is regulated, which influences the thrust generated by the wings. The only difference is that in the proposed mechanism the wing root is translated, instead of rotated.

A larger difference is found for the pitch and yaw deflections. Even though the wing roots are deflected in a similar way as on the Nano Hummingbird and the KUBeetle, the effect of these deflections on the wings is different. Instead of changing the wing tension during the flap cycle (which is very effective at large flapping amplitudes), the airflow around the wings is vectored, and thereby the thrust generated by the wing is vectored. As a result, the pitch and yaw moments generated by the proposed mechanism are opposite to those generated by the mechanisms of the Nano Hummingbird and the KUBeetle. The magnitude and direction of the pitch moment depends on the location of the center of gravity.

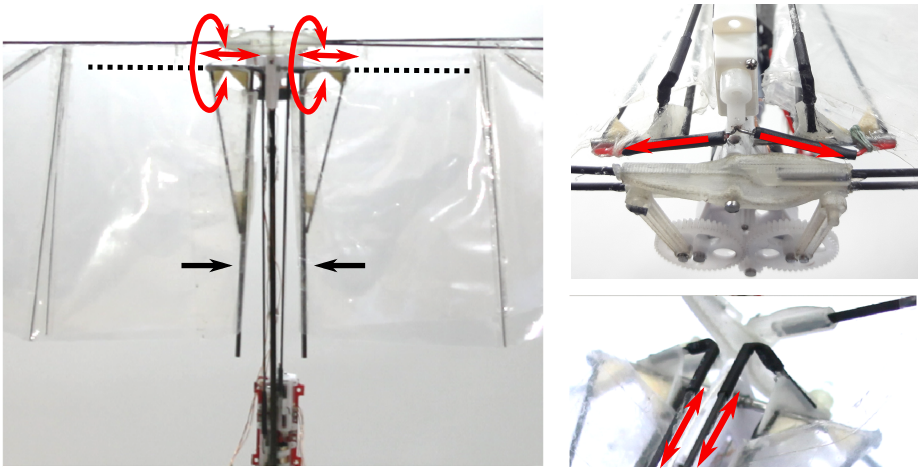

Figure 6.3: **Left**: Image of the control mechanism as implemented on the DelFly. Two carbon rods function as the wing roots which are placed inside the left and right wings (indicated by black arrows). The rods can rotate and translate (red arrows) around hinges (dashed lines) which are close to the leading edges of the wings. Two linear servos (visible at the bottom of the left image) actuate the hinges via carbon push rods. **Bottom-right**: Detailed image of the hinges, showing the individual motions of the push rods (red arrows) for the rotational motions, and their connection with the wing root hinges. **Top-right**: Detailed image of the hinges, showing the individual motions of the push rods (red arrows) for the translational motions, and their connection with the wing root hinges.

When deflecting the wing roots for pitch and yaw inputs, the tension of the wings is not affected. This is visualized by Fig. 6.5. The left image shows that if the left wing root is deflected, the left wing is tilted as a whole, and thereby changes the thrust vector. The left image shows the situation when the wings are almost closed. The right image shows the situation when the wings are opened. This shows that the shape of the wing deforms during the flapping cycle: the upper part of the left wing becomes smaller, and the lower part of the left wing becomes larger. Because of this effect, the tension in the wing does not change during the flapping cycle. In other words, the wings are not stretched by the mechanism. A benefit of this property is that very large root deflections can be realized.

As Fig. 6.4 clearly shows, the benefit of the proposed wing control system is its simplicity. Both wing roots have two degrees of freedom, rotating and sliding, using a single joint (see right image of Fig. 6.3). The joint is made of thin walled steel tubes that form a clearance fit with minimal play. As these tubes are made from syringe needles, the tube walls are very smooth and can easily rotate and slide with respect to each other. An extra supporting carbon rod is used to obtain a stiff connection between the joint and the wing root, forming a triangular structure. Furthermore, the joints of this structure are reinforced by balsa wood.

Apart from these components, only servos and pushrods are needed for actuation. As Fig. 6.3 shows (bottom-right), two pushrods are connected to the hinges for the rotational motions, driven by two servos. A third servo can be added for translating the hinges. This is shown in the top-right image of this figure.

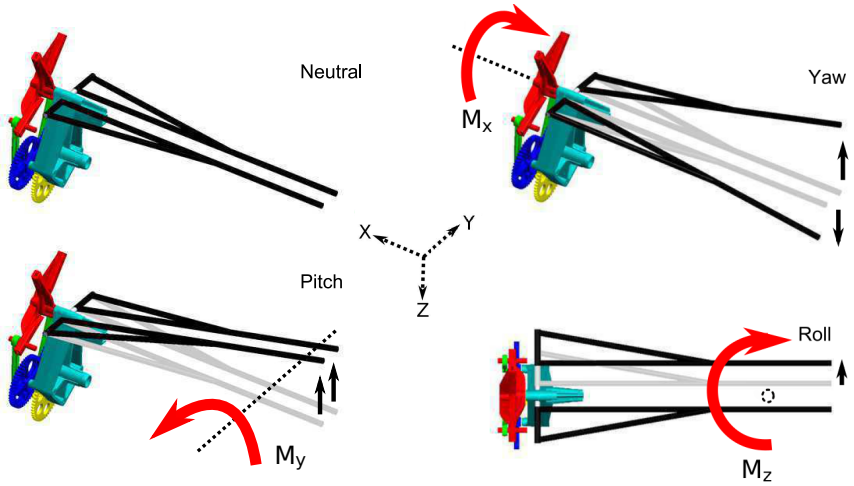

Figure 6.4: Visualization of the proposed control mechanism for different control inputs. The small (black) arrows indicate the direction in which the wing roots are deflected. The large (red) arrows indicate the control moment generated by the control inputs around the body axes which are indicated by the dashed lines. The names of the axes (pitch, yaw and roll) correspond to the situation in hover flight.

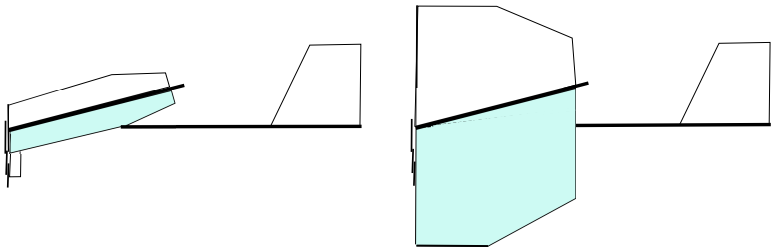

Figure 6.5: Schematic illustration of the effect of wing root deflection on the wing shape. A side view of the DelFly is presented with the wings almost closed (left) and fully opened (right). In both cases, the wing root of the left wing is deflected upwards. The left image shows that the wing root deflection makes the wing rotate upwards as a whole and the root is parallel to the center line of the wing. The right image shows that the wing root has shifted upwards with respect to the wing center line (dashed line). During flapping, the wing slides over the wing root every cycle whereby the top and bottom parts of the wing change their shape continuously.

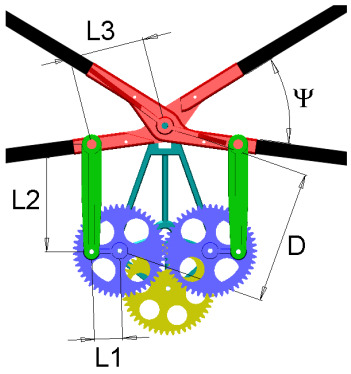

Figure 6.6: Schematic drawing of the original DelFly II flapping mechanism, which includes a dihedral angle of $12°$. The dimensions of the four-bar mechanism are defined, as well as the wing flap angle.

6.4. Flapping Mechanism

The flapping mechanism as used in this study is fully based on the original mechanism of the DelFly II [20, 27]. The platform has four wings that are arranged in a biplane configuration. The wings are coupled crosswise, as illustrated in Fig. 6.6. As a result the top and bottom wings flap in anti-phase which is beneficial for limiting platform vibrations. Furthermore, the system benefits from a clap-and-peel [28] effect which is present at the end of the down-stroke and the start of the upstroke, where the two wings meet. The flapping mechanism consists of two symmetric four-bar mechanisms that are coupled. As a result, the wings flap fully synchronized, which is also beneficial for limiting platform vibrations. The mechanism is driven by a single DC motor. Its speed is reduced by a ratio of $1:21.33$ via a two stage gearbox.

In this study, an adapted version of the flapping mechanism is used, which shown in Fig. 6.7. Originally the wing hinges have a dihedral angle of $12°$, which enhances yaw stability in slow forward flight. This dihedral angle is undesirable in the current study since the wings are used for control purposes. The dihedral angle would result in asymmetry and coupling effects of yaw, pitch and roll inputs. Furthermore, it would limit the maximum deflections of the control mechanism which uses angle of attack modulation.

The definition of the characteristic dimensions of the four-bar flapping mechanism are indicated in Fig. 6.6. These dimensions, together with the stroke amplitude, are specified in Table 6.1 for both the original and current design. Because $L3$ has been made longer to maintain smooth rotations, the stroke amplitude Ψ_{max} is smaller in the current design. This influences the relation between motor speed and generated thrust, and the overall efficiency of the flapping system.

The plastic components of the original flapping mechanism are injection molded. For the current mechanism, the pushrods and the wing hinges have been redesigned and are 3D-printed, using a UV-curable acrylic resin (Shapeways Frosted

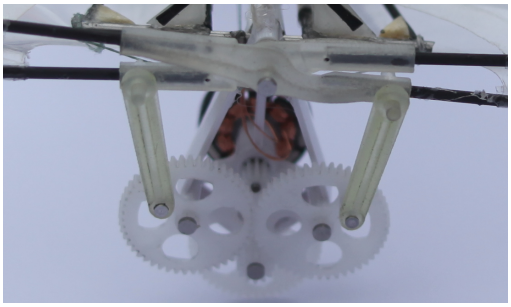

Figure 6.7: Image of the flapping mechanism as used in this study, which does not include a dihedral angle.

Table 6.1: Comparison of flapping mechanism parameters (as in Fig. 6.6) for original and proposed design

Design	$L_1[mm]$	$L_2[mm]$	$L_3[mm]$	$D[mm]$	$\Psi_{max}[^\circ]$
original	4.55	17.22	12.07	21.49	88
proposed	4.55	15.09	13.08	21.49	83

Ultra Detail).

The wings in the current system are similar to the wings in the original system, with a wingspan of $28\ cm$, mean chord length of $8\ cm$ and aspect ratio of 1.75. They are made of Mylar foil and carbon rods. The leading edges are attached to the wing hinges.

6.5. Static performance tests

The forces and moments generated by the proposed flapping system were analysed using a static test setup. In this setup, the whole mechanism including wings is attached to an ATI Nano 17 (force resolution 3.125 mN, moment resolution 15.625 mNmm) transducer that measures 6-DOF forces and moments. A combination of a hall sensor switch and a magnet in the flapping mechanism was used to measure the flapping frequency.

The effect of giving a pure pitch input (symmetric deflection of wing roots, Fig. 6.4 second from top) is shown in Fig. 6.8. The results show that the pitching moment M_Y can be approximated by a linear fit. Furthermore, the results show that the effect on the other two moments is significantly smaller. For the tested range of inputs ($\pm 25\ deg$) a maximum variation in pitch moment of approximately $6\ Nmm$ is observed. The measured magnitude of the pitching moment depends on the location of the centre of gravity. In this test, the location of the centre of gravity corresponds with Fig. 6.15.

The effect of giving a pure yaw input (differential deflection of wing roots) is shown in Fig. 6.9. In this case a linear variation in yawing moment M_X is observed,

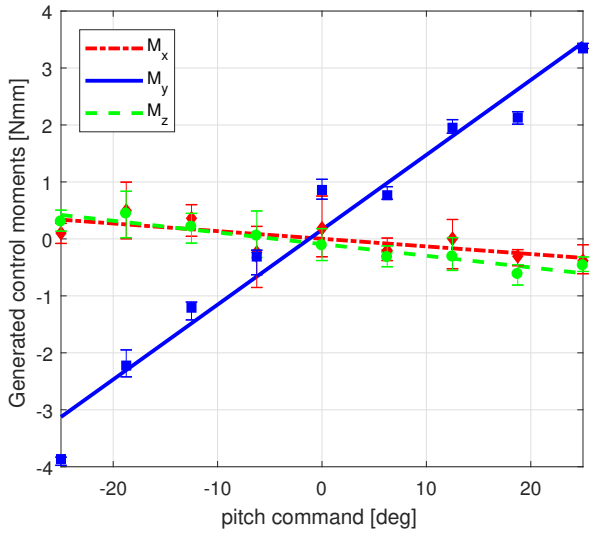

Figure 6.8: Generated moments around all three axis for a range of pitch inputs. A positive pitch command means that both wings are deflected upwards. The flapping frequency in this test is $\sim 15\ Hz$. All measurements are time-averaged over 29 flapping cycles. Linear fits through these measurements are shown. The error bars show the minimum and maximum time-averaged measurement over a single flapping cycle.

as is expected. The effect on the other two axis is again significantly smaller. For the tested range of inputs ($\pm 25\ deg$) a variation in pitch moment of approximately $5\ Nmm$ is observed.

A coupling effect on the Z-axis (roll) does occur when a combination of pitch and yaw commands is given. This is visible from the results in Fig. 6.10. Note that the maximum input commands are only 50% of the maximum pitch and yaw inputs, but that the sum of these inputs also covers the combination with maximum coupling effect. The results do not show perfect symmetry, but clearly show the dependency on the combination of the two inputs. The variation in roll moment is approximately $1.5\ Nmm$, and clearly influences the maximum yawing inputs that can be given. For comparison, the result of the combinations of pitch and yaw commands on the Y-axis (pitch) is shown in Fig. 6.11. This shows that the generated pitch control moment mainly depends on the pitch command, and is significantly less dependent on the yaw command, as is expected.

The effect of giving a pure roll input is shown in Fig. 6.12. The command is defined as the translational displacement of one of the two wing roots as explained in Fig. 6.4. A positive command means that the right wing root is shifted outwards. As expected, the roll input mainly has an effect on roll moment M_Z. In this test, the maximum variation in M_z that can be generated is about $1.4\ Nmm$. Note in Fig. 6.10 that a comparable variation in M_Z is measured as the coupling effect between pitch and yaw inputs (Fig. 6.10). This indicates that the stability margin for roll will be

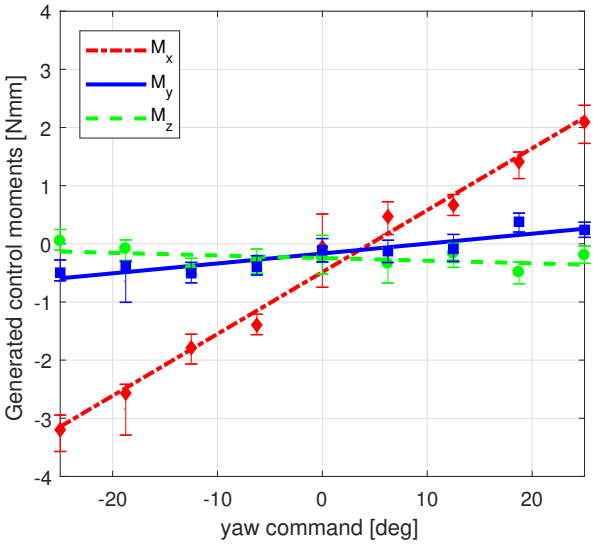

Figure 6.9: Generated moments around all three axis for a pure yaw input. A positive yaw command means that the left wing is deflected downwards, and the right wing is deflected upwards. The flapping frequency in this test is $\sim 15\ Hz$. All measurements are time-averaged over 29 flapping cycles. Linear fits through these measurements are shown. The error bars show the minimum and maximum time-averaged measurement over a single flapping cycle.

small under certain conditions. Note that the roll input also has an effect on the yaw moment M_X which is caused by an imperfect differential deflection of the left and right wing roots.

6.6. Attitude Estimation and Control

A lightweight custom-made autopilot board of 1 gram is used to enable attitude estimation and control. Fig. 6.13 shows the board attached to the vehicle. It features an Atmel ATmega328P RISC-based 8-bit microcontroller that receives input from multiple sensors: 6-axis gyroscope and accelerometer readings from an InvenSense MPU9150, 3-axis magnetometer readings from a Honeywell HMC5883L, and barometric pressure readings from a Bosch BMP180. The autopilot board also features a transceiver for two-way communication (remote control and telemetry). The microcontroller runs an attitude estimation and control algorithm based on these inputs and controls the vehicle via the integrated electronic speed controller and external servos.

Initial tests reveal that the autopilot board cannot be attached to the main structure of the vehicle. The mechanical vibrations from the system contain frequencies that exceed the Nyquist frequency of the accelerometer sensors. Therefore, the board is attached to a substructure that includes foam parts for mechanical damping. The battery is also attached on it to lower the natural frequencies of the

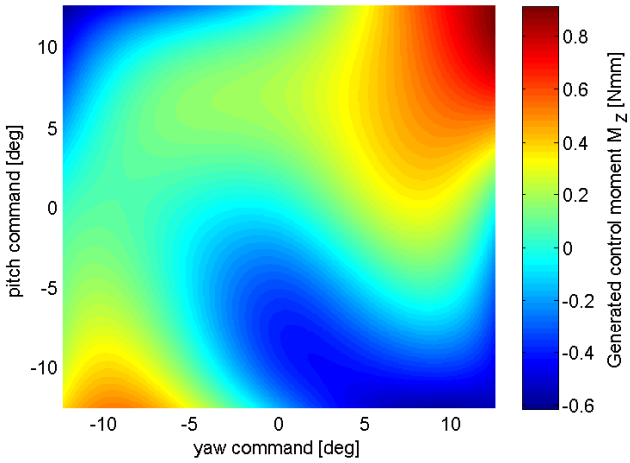

Figure 6.10: Generated roll moments (M_Z) as a coupling effect from combinations of pitch and yaw commands. The commands for both inputs ranges to a maximum of $\%50$, such that the summed inputs never exceed $\%100$. The flapping frequency in this test is $\sim 15\ Hz$.

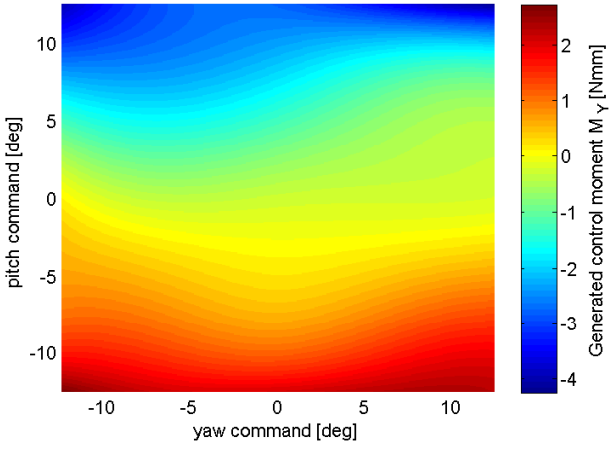

Figure 6.11: Generated pitch moments (M_Y) for combinations of pitch and yaw commands. The commands for both inputs ranges to a maximum of $\%50$, such that the summed inputs never exceed $\%100$. The flapping frequency in this test is $\sim 15\ Hz$.

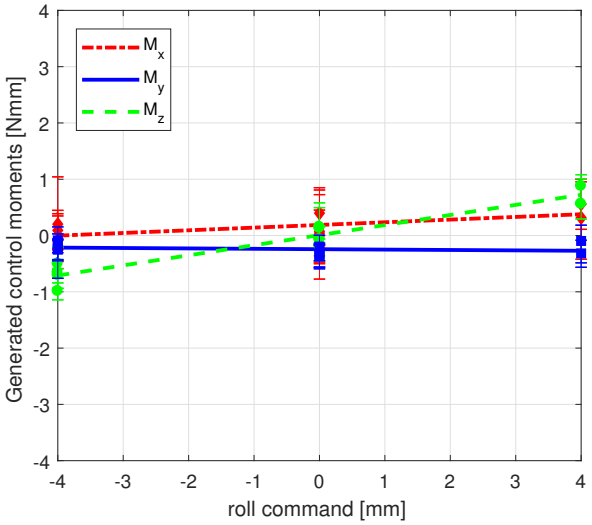

Figure 6.12: Generated moments around all three axis for a pure roll input. A positive roll command means that the right wing root is shifted outwards. The flapping frequency in this test is $\sim 15\ Hz$. All measurements are time-averaged over 29 flapping cycles. Linear fits through these measurements are shown. The error bars show the minimum and maximum time-averaged measurement over a single flapping cycle.

substructure. This can also be seen in Fig. 6.13.

To show the importance of this approach, a comparative test was performed to show the effect of using the substructure. Fig. 6.14 shows unfiltered measured X-axis accelerations from test flights with and without the damping system. The sampling frequency is $1kHz$ in both tests. The result for the undamped case shows noisy measurements that contain high frequency components exceeding the Nyquist frequency. The result for the test with damping system shows a smooth and periodic measurement which is dominated by the flapping frequency ($\sim 14Hz$ in this test).

6.6.1. Attitude estimation

The attitude is represented by Euler angles: roll (ϕ), pitch (θ) and yaw (ψ). The accelerometer measurement $\mathbf{a} = [a_x, a_y, a_z]^T$ is used to estimate the roll and pitch angles; the magnetometer measurement $\mathbf{m} = [m_x, m_y, m_z]^T$ is incorporated to estimate the yaw angle. For the definition of the Euler angles, the $2-1-3$ rotation sequence (pitch-roll-yaw) is used. The angles are defined as:

$$\begin{aligned}\tilde{\theta} &= atan(\frac{-a_x}{a_z}) \\ \tilde{\phi} &= atan(\frac{a_y}{\sqrt{a_x^2+a_z^2}}) \\ \tilde{\psi} &= atan(\frac{\sin(\tilde{\phi})\sin(\tilde{\theta})m_x+\cos(\tilde{\phi})m_y+\sin(\tilde{\phi})\cos(\tilde{\theta})m_z}{\cos(\tilde{\theta})m_x+\sin(\tilde{\theta})m_z})\end{aligned} \tag{6.1}$$

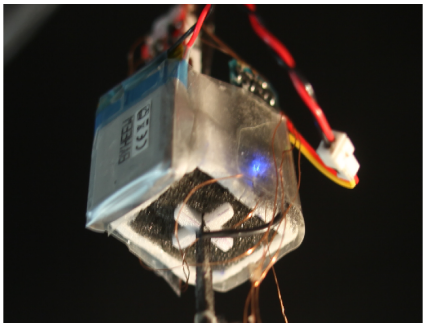

Figure 6.13: Damping structure as used to reduce the effect of mechanical vibrations on the sensors of the autopilot. It consists of a (white) rigid foam box, with (black) damping foam inside which encloses the fuslage. Both the battery and the autopilot board are firmly attached to the foam box such that they form a rigid mass.

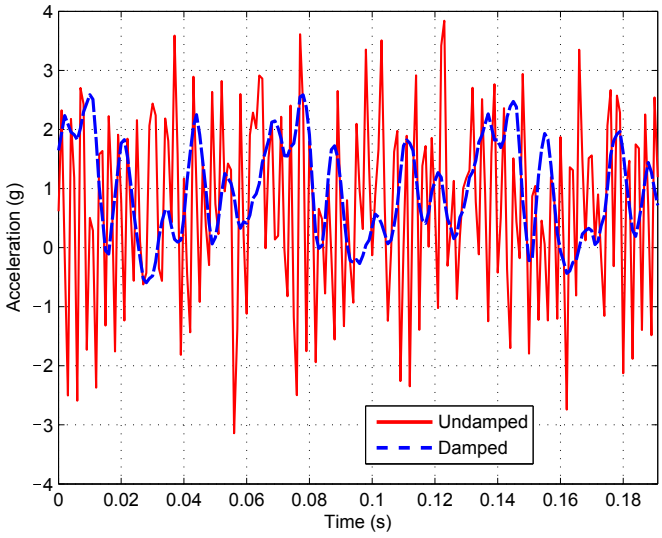

Figure 6.14: The effect of mechanically damping the autopilot board. The results show the X-axis accelerometer measurements for the undamped and the damped system at a large pitch angle ($\sim 80°$).

In this equation, $\widetilde{\theta}$, $\widetilde{\phi}$ and $\widetilde{\psi}$ are estimates of the pitch, roll and yaw angle respectively. These are based on the accelerometer measurement **a** and magnetometer measurement **m**. This definition of the Euler angles ensures that the pitch angle is defined also for high pitch angles ($> 90°$) that occur during hover. Furthermore, the chosen rotation sequence also leads to more stable estimations of the roll and pitch angle. The reason for this is that the DelFly (normally) flies with small roll angles, which means that the gravitational force is expected to be predominately present on the X axis, around hover, and also on the Z axis, in slow forward flight. From Equation 6.1 it can be seen that the roll and pitch angle estimates are not sensitive to noise measured on the Y axis. The CORDIC algorithm [29] is used on board to calculate all the necessary trigonometric functions. It is important to mention that in this definition of the roll angle, the relation between the roll angle and the body axes depends on the pitch angle. The definition of the roll and yaw body axes in Fig. 6.4 corresponds to the hover condition (90° pitch).

Accelerometer data filtering

As is shown in Fig. 6.14 the accelerometer measurements are mainly disturbed by the periodic flapping motion of the wings, but also seem to be influenced by other harmonics. For this reason, the accelerometer measurements are individually filtered per axis using a moving average filter. The window of the filter is defined by the flapping frequency such that it matches the period of the flapping cycle.

Accelerometer and gyroscope fusion

The moving average filter results in smoother accelerometer measurements that enhance the angle estimates, but at the cost of adding delay to the signal. These estimates are therefore fused with the faster gyroscope measurements $\mathbf{g} = \left[g_x, g_y, g_z\right]^T$ using a first-order complementary filter. The function of the filter for the pitch angle is given by:

$$\hat{\theta} = \widetilde{\theta}(1-\tau) + (\hat{\theta} + (\cos(\hat{\phi})g_y + \cos(\hat{\theta})\sin(\hat{\phi})g_z)\Delta t)\tau \tag{6.2}$$

where $\hat{\theta}$ is the filtered pitch estimate, $\widetilde{\theta}$ is the estimated pitch angle based on Eq. 6.1, Δt is the time step and τ is a weight factor which is set at 0.9. The roll and yaw angle are filtered in the same way using their corresponding components of gyroscope measurements.

6.2.2. Attitude control

The estimated attitude angles are used as feedback in the attitude control loop. In this study, the focus is on hover and slow forward flight. Therefore, the Euler angle estimates are used in a straightforward manner to compute the error with the reference angles. It is thus assumed that the definition of pitch, roll and yaw in Eqs. 6.1 and 6.2 corresponds to the definitions in Fig. 6.4. PD-control is used to achieve stabilization. The trim settings for all angles are manually tuned.

6.7. Characteristics of platform with tail

The first DelFly design that is proposed in this study includes a tail. An active rudder surface is used in this design because the type of servo used for roll is not powerful enough to implement it for wing-based roll control. A visual overview of this DelFly design is shown in Fig. 6.15. The figure indicates the locations of the main components, as well as the location of the centre of gravity. Apart from the control and flapping mechanisms, the main components of the system are the autopilot board, the servos and the battery. Detailed information about the autopilot board is discussed in Sec. 6.6. A SuperMicro Systems - Double Linear Servo ($2.4\ g$) is used to actuate the left and right wing roots individually. The rudder is actuated by a Sub Micro LZ servo ($0.5\ g$). A $180\ mAh$ LiPo battery is used in the experiments. A mass breakdown of the total system is given in Table 6.2. Furthermore, an overview of vehicle characteristics is presented in Table 6.3.

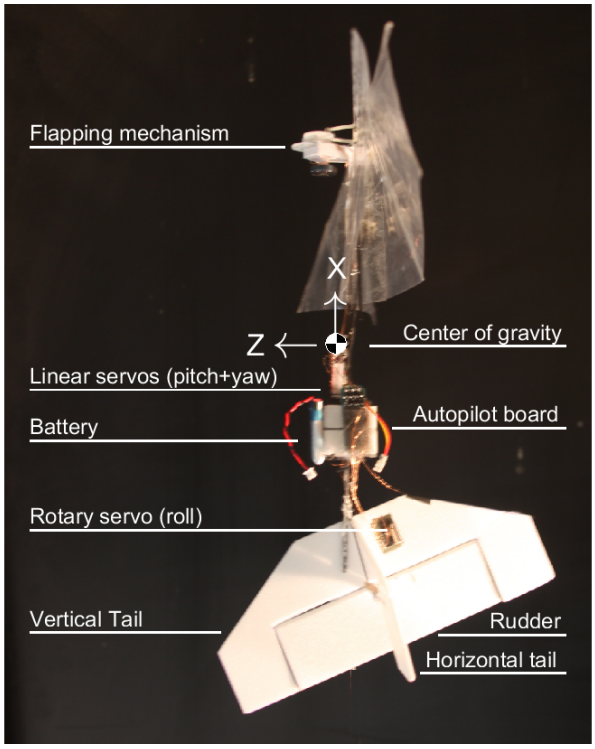

Figure 6.15: Overview of the DelFly II with wing control mechanism. The image shows the attitude of the vehicle in hover (90 deg pitch), which corresponds to the attitude during avoidance. This design is used for the flight tests described in Sec. 6.8 and contains an active rudder but no active elevator. Pitch control is controlled solely by the wing control mechanism. The location of the center of gravity is indicated, as well as the definition of the (right-handed) body-axes. A detailed image of the flapping mechanism is shown in Fig. 6.7. A detailed image of the control mechanism is shown in Fig. 6.3. The damping structure attachment of the battery and the autopilot board is more clearly visualized in Fig. 6.13.

Table 6.2: Mass breakdown of design with tail

Component	Mass (g)
Motor	1.0
Flapping mechanism	2.7
Wings (2)	1.4
Control mechanism	1.5
Servos(3)	2.9
Autopilot board	1.0
Fuselage	1.0
Damping structure	0.8
Tail	2.3
Battery	5.5
Wiring and glue	1.0
Total mass	21.1

Table 6.3: Design and performance characteristics

Total mass	21.1	g
Wing span	28	cm
Wing area	224	cm^2
Wing aspect ratio	8	-
Flap amplitude	83	deg
Flight Speeds	0-0.5	m/s
Minimum turn radius	0.1	m

A second detail that is to be noted is the negative pitch angle of the horizontal tail section. In the current design, this is necessary to allow stable slow forward flight at high pitch angles, while previous DelFly designs did not need this adjustment. This is due to the wing dihedral angle which is removed in the current design. As a result, the thrust vector produced by the wings is almost collinear with the body X-axis (see Fig. 6.15). In the original design including the dihedral angle, the thrust vector acts further away from the body, which creates an additional pitch down moment (around the Y-axis). The interaction between this pitching moment and the aerodynamic moments acting on the wing and the tail results in a stable forward flight pitch angle. For the original design, the pitch angle is between 70° and close to 90° (almost hover). For the current design this stable pitch angle regime is smaller, and close to 90°, which restricts forward speed. The negative pitch angle of the horizontal tail section in the current design, in combination with a negative pitch angle of the wings, results in a stable forward flight motion. In this case, the pitch angle is still close to 90°, but the wing and tail vectoring results in an additional forward force. Another method that will be explored in the future is to move the center of gravity further up along the direction of the body Z-axis by repositioning heavy components.

6.8. Flight Tests of platform with tail

Flight tests were performed to demonstrate the effectiveness and performance of the proposed wing control mechanism in the configuration from Fig. 6.15. A distinction is made between two different flight conditions: hovering and slow forward flight. The hovering condition was previously tested on a DelFly II using elevator and aileron surfaces [19]. In these tests, it was shown that the attitude control loop enabled the stabilization of the pitch angle. However, the tests showed that the effectiveness of the ailerons is insufficient to control the yaw angle during hovering. The hovering tests in the current study are performed to show that the proposed wing control mechanism allows both pitch and heading control.

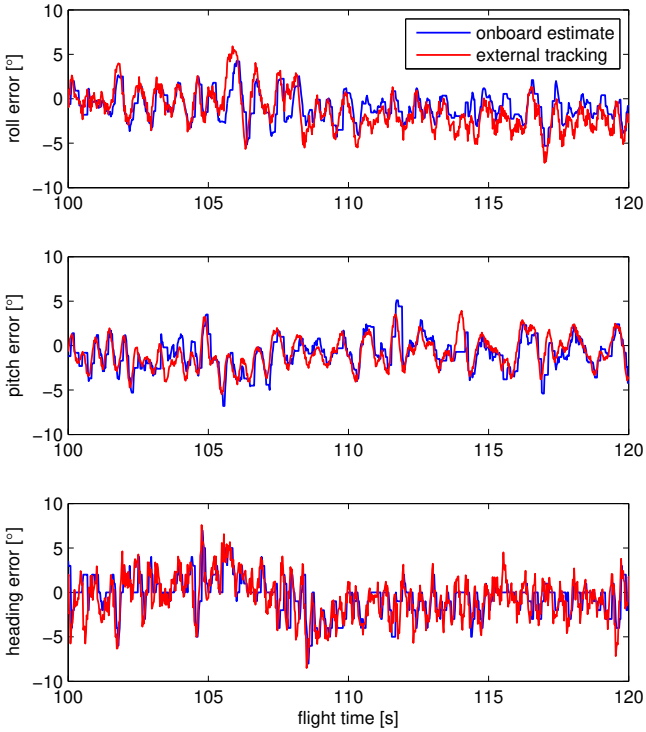

Figure 6.16: Hovering test results showing the attitude tracking performance of the vehicle for static reference angles (pitch 90° and roll 0°) such that the vehicle hovers. Both onboard attitude estimates and externally tracked attitude values are shown for comparison, without further filtering or smoothing. The onboard estimates are obtained using the filtering method described in Sec. 6.6, which is based on a moving average filter. The number of samples of this filter is dynamic, as this number corresponds to the period of the flapping cycle.

The slow forward flight condition is also tested as this is an important flight condition for the DelFly II when performing autonomous obstacle avoidance and

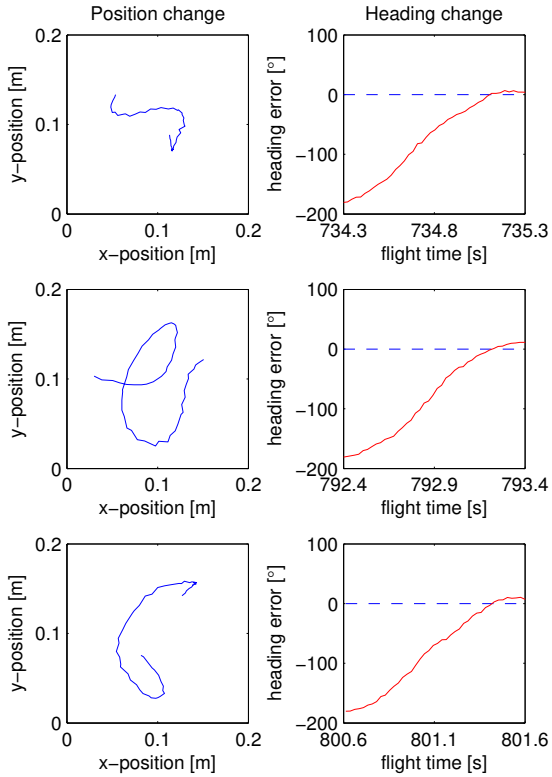

Figure 6.17: Lateral position (x,y) and heading responses of the vehicle during three 180-turns. The turns were performed while starting from the hovering condition.

navigation in indoor environments. In Chapter 2 and 3 aileron surfaces were used to enable smooth turns that allow the vehicle to perform visual navigation at the same time. However, situations occur where the vehicle needs to turn around quickly to avoid collisions with obstacles that were not detected sufficiently ahead. The flight tests in this study show that in such a case the vehicle is capable of smoothly transitioning from forward flight to hover, and that it can perform a fast turn to instantly reverse its heading.

The tests are performed in a 10×10×7 m flight arena equipped with an OptiTrack Motion tracking system consisting of 24 Flex13 cameras. This data is used for post-flight analysis of the flight performance.

Results from the hovering test are shown in Fig. 6.16. The tracking performance of the roll, pitch and yaw angles are shown for 20 seconds of flight. It can be observed that the onboard attitude estimates correlate well with the externally observed attitude angles. Furthermore, the results show that during this test the

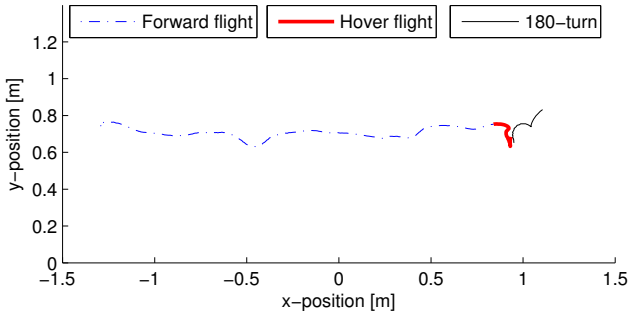

Figure 6.18: Flight track in the horizontal plane (top view) of the vehicle during a stop-and-turn maneuver. Initially the vehicle flies forward, and is then commanded to transition to hover attitude. Once stable hover has been reached the vehicle is instructed to perform a 180-turn.

maximum attitude errors for all three attitude angles are in the order of $\pm 5^{\circ}$. This clearly illustrates the effectiveness of the wing control mechanism in controlling the pitch and yaw angles.

Attitude and position data results from fast 180° turns during hover are shown in Fig. 6.17. The results show that the vehicle is able to smoothly turn around in one second while requiring a turn radius of less than $0.1\ m$. From the position plots in the figures it can be observed that the response of the vehicle during the turn is different each time. While the heading responses show similar results, the lateral motion of the vehicle is more random.

In Fig. 6.18, the flight path of the vehicle is shown for a test where the vehicle initially flies with a forward speed of $29\ cm/s$ and is then commanded to hover and turn 180 degrees using its wing control mechanism. The figure shows the horizontal motion of the vehicle during these three different phases. This result shows that the vehicle is able to perform a stop-and-turn maneuver during slow forward flight that requires the vehicle to move over a distance of less than $25\ cm$. The maneuver takes $2.5\ s$. The attitude tracking performance of the vehicle during the forward flight phase of this test is shown in Fig. 6.19. These results show that the attitude tracking performance of roll, pitch and yaw are significantly worse compared to hovering flight. Since the tracking performance is worse for all three attitude angles it is not possible to identify a main reason for this result. However, the poor pitch stability in this test seems to point out that the current tail configuration, as discussed in Sec. 6.7, does not provide the same forward flight stability benefits that were observed in flight tests with the DelFly Explorer in Sec. 5.6.

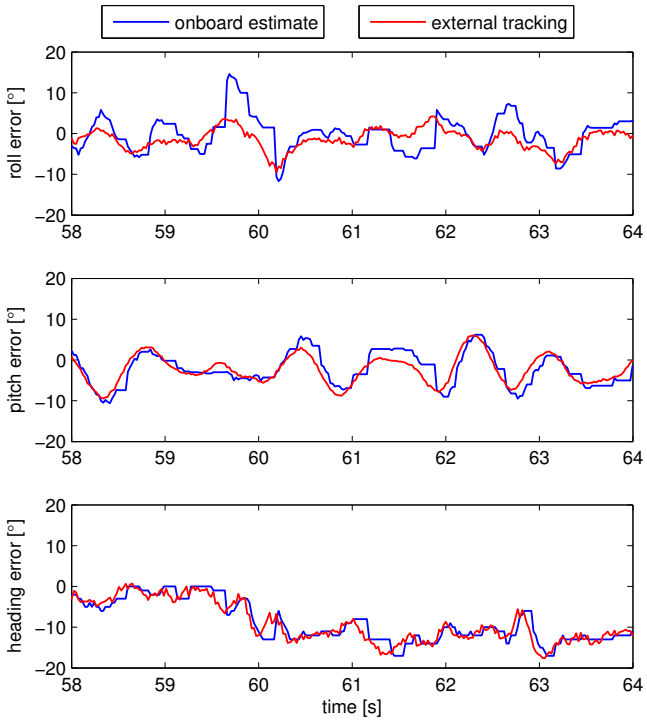

Figure 6.19: Attitude tracking performance of the vehicle in slow forward flight. Both onboard attitude estimates and externally tracked attitude values are shown for comparison.

6.9. Characteristics of platform without tail

This section presents the proposed mechanism without the additional actuator for roll. The main difference of this design, shown in Fig. 6.20, is that it has no tail. This has several consequences for the total mass of the system, which is reduced to 19.7 g. A mass breakdown of this design can be found in Table 6.4. A significant amount of mass is saved by removing the tail (2.3 g). As a consequence, the length of the carbon fuselage rod can be reduced, saving 0.5 g. Instead of the two linear servos, two HobbyKing HK-5330S Ultra-Micro Digital Servos (1.7 g) are used in combination with the Sub Micro LZ servo (0.5 g). The damping structure used in the first design is replaced by a more minimalistic version that uses only a foam layer around the fuselage rod that prevents direct contact between the fuselage and the battery-autopilot combination. This solution is visualized in the top right image in Fig. 6.20. In this design, the wings are heavier, 1.3 g each. This is the result of adding an extra layer of tape on the middle section of the wing. This prevents the carbon wing root rod from wearing the mylar foil. Replacing this wing section with only tape to save weight is not an option, as the inelastic property of the mylar foil is needed to maintain the required wing tension.

Table 6.4: Mass breakdown of tailless design

Component	Mass (g)
Motor	1.0
Flapping mechanism	2.7
Wings (2)	2.5
Control mechanism	1.5
Servos(3)	3.9
Autopilot board	1.0
Fuselage	0.5
Damping structure	0.1
Battery	5.5
Wiring and glue	1.0
Total mass	19.7

The two HK-5330S servos are used in this design for their fast response rate. According to the manufacturers specifications: 60° in 0.04 s at 4.2 V. In comparison, the linear servos used in the design with the tail have a specified speed of 0.12 s at 4.2 V. The rotary servos are also more powerful, which also influences the response time of the mechanism in flight. For stable flight it is required to control the pitch and roll angles. The bottom center image in Fig. 6.20 shows how one HK-5330S servo is used to actuate the roll mechanism. The bottom left image in Fig. 6.20 shows how the second HK-5330S servo is implemented to control the pitch mechanism of both wing roots. The small Sub Micro LZ servo is integrated in this mechanism. This servo is not able to deliver the force and speed required for pitch stabilization, but forms a lightweight solution to introduce an offset between the left and right wing roots for enabling yaw control. To allow more aggressive control on the pitch and yaw axes, the two wing roots can also be actuated by two individual HK-5330S servos. This adds approximately 1 g to the total mass of the system.

6.10. Flight Tests of platform without tail

Flight tests with the system in Fig. 6.20 were performed to test whether the proposed wing control mechanism allows the flapping wing vehicle to perform stable flight without having a tail. Flight times of up to 5 minutes of continuous hovering flight have been realized. Longer flight times are expected to be possible in terms of battery capacity, but are hindered by instabilities of the system. Instability occurs due to disturbing wind conditions, and are mainly an issue for roll stability. As the results from Sec. 6.5 indicate, the proposed mechanism is the least effective in roll.

Pitch instability also forms an issue using this wing control mechanism. The reason for this is that the location of the center of gravity (CoG) influences the pitch control effectiveness of the mechanism. This effectiveness increases if the CoG is located further away from the leading edge of the wing. At the same time, the CoG is ideally located at the aerodynamic center (AC) of the wing, such that flight speed has no influence on the pitching moment coefficient of the wing. This

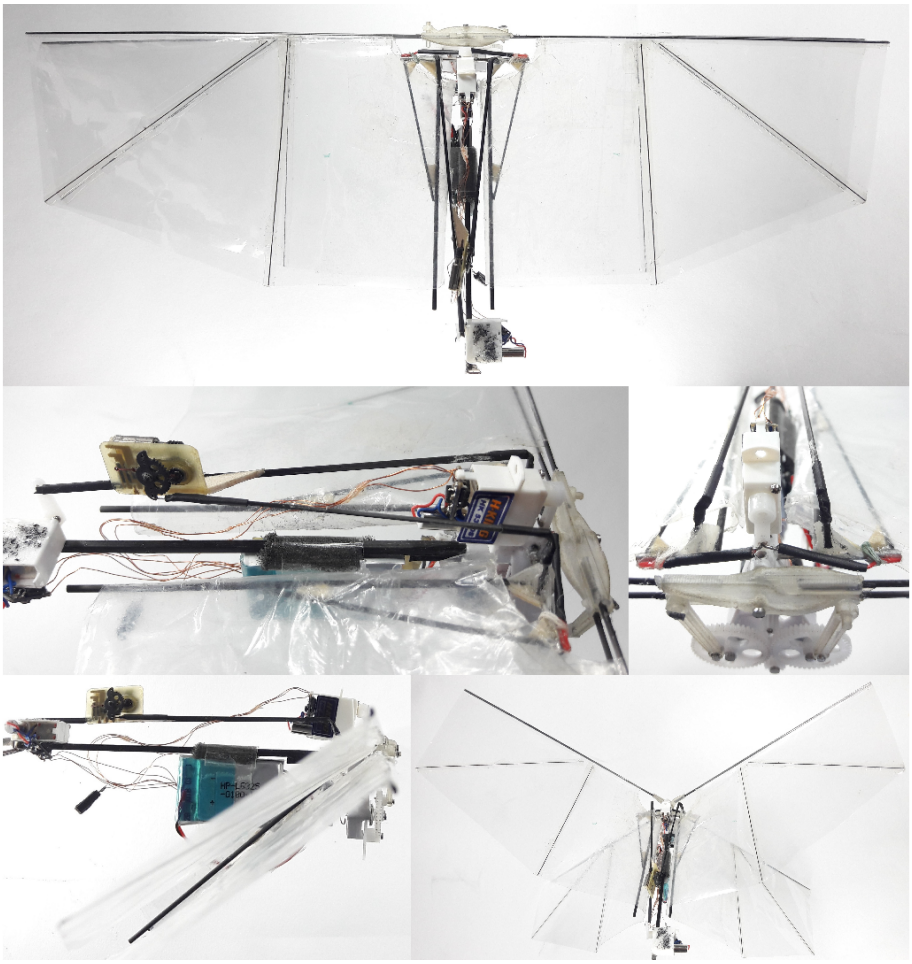

Figure 6.20: Tailless DelFly design that uses the proposed wing control mechanism. **Top**: Top view showing a clear overview of the vehicle and illustrating that all components are compactly concentrated around the fuselage. **Middle left**: Detailed view showing the mechanism for pitch and yaw control. **Middle right**: Detailed view showing the mechanism for roll control. **Bottom left**: Side view of the vehicle to clearly show where all components are placed. Note how the foam is used in the connection between the battery and the fuselage. **Bottom right**: Overview of the vehicles' appearance when the wings are opened.

forms a conflict, because the pitch control effectiveness of the mechanism is too small when the CoG is located at the AC. Therefore the CoG is located further aft in the current vehicle configuration. This allows only very limited flight speeds, as pitch instabilities occur otherwise. The vehicle can thus be trimmed for hovering flight and also for slow forward flight, as can be seen in the first two videos of the associated playlist[1]. Furthermore, the roll control mechanism allows the vehicle to

[1] https://www.youtube.com/playlist?list=PL_KSX9GOn2P-6D1pNf1tAAcs586EQeBCg

perform lateral maneuvers.

More detailed tests and design improvements are required to explore the stability limits of the system. The effectiveness of the wing control mechanism can be improved by several design changes. Current experiments also indicate that even a small tail can improve the vehicle stability considerably (third video of the playlist). A combination of a right-sized tail and the proposed wing control mechanism is therefore expected to deliver a high reliability with the ability to perform aggressive maneuvers, especially in pitch and yaw.

6.11. Conclusions

In this study, a wing control mechanism is presented for a flapping wing MAV having two pairs of wings in a biplane configuration. The mechanism serves as a lightweight control solution that allows for fast and large wing root deflections. Both symmetric and asymmetric wing root deflections are possible by which pitch and yaw moments can be created. Furthermore, the proposed mechanism allows for lateral translations of the wing roots by which the tension of the wings can be adapted, enabling roll moments. Static tests show that pitch and yaw moments of sufficient magnitude can be created for hover stabilization. The magnitude of the produced roll moments is in the same order as the maximum roll moments that are produced by the coupling effect of combined pitch and yaw commands. External disturbances determine the effective roll moments that are created. Real flight tests are presented that show that stable hover flight is realized in combination with a tail and that this system is also able to perform fast turns in this condition. Flight test videos with a tailless configuration are also discussed, demonstrating hovering and slow forward flight, and also sideways flying. Other experiments indicate that aggressive pitch and yaw maneuvers can be combined in a configuration with a tail.

References

[1] C. De Wagter, D. Dokter, G. C. H. E. De Croon, and B. D. W. Remes, *Multi-lifting-device uav autonomous flight at any transition percentage,* Proceeding of: EuroGNC **2013** (2013).

[2] A. F. Şenkul and E. Altuğ, *System design of a novel tilt-roll rotor quadrotor uav,* Journal of Intelligent & Robotic Systems **84**, 575 (2016).

[3] K. Wang, Y. Ke, and B. M. Chen, *Autonomous reconfigurable hybrid tail-sitter uav u-lion,* Science China Information Sciences **60**, 033201 (2017).

[4] C. De Wagter, R. Ruijsink, E. Smeur, K. van Hecke, F. van Tienen, E. van der Horst, and B. Remes, *Design, control and visual navigation of the delftacopter,* arXiv preprint arXiv:1701.00860 (2017).

[5] W. Send, M. Fischer, K. Jebens, R. Mugrauer, A. Nagarathinam, and F. Scharstein, *Artificial hinged-wing bird with active torsion and partially linear*

kinematics, in *28th Congress of the International Council of the Aeronautical Sciences* (2012) pp. 23–28.

[6] L. Roberts, H. A. Bruck, and S. K. Gupta, *Autonomus loitering control for a flapping wing miniature aerial vehicle with independent wing control,* in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (American Society of Mechanical Engineers, 2014) pp. V05AT08A013–V05AT08A013.

[7] N. Gaissert, R. Mugrauer, G. Mugrauer, A. Jebens, K. Jebens, and E. M. Knubben, *Inventing a micro aerial vehicle inspired by the mechanics of dragonfly flight,* in *Towards Autonomous Robotic Systems* (Springer, 2013) pp. 90–100.

[8] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov, *Development of the Nano Hummingbird: A Tailless flapping wing micro air vehicle.* in *50th AIAA Aerospace Science Meeting* (2012) pp. 6–12.

[9] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, *Controlled Flight of a Biologically Inspired, Insect-Scale Robot.* Science **340**, 603 (2013).

[10] M. Karásek, A. Hua, Y. Nan, M. Lalami, and A. Preumont, *Pitch and roll control mechanism for a hovering flapping wing mav,* International Journal of Micro Air Vehicles **6**, 253 (2014).

[11] D. Coleman, M. Benedict, and I. Chopra, *Design, development and flight testing of robotic hummingbird,* in *71st Annual Forum of the American Helicopter Society* (2015).

[12] H. V. Phan, T. Kang, and H. C. Park, *Design and stable flight of a 21 g insect-like tailless flapping wing micro air vehicle with angular rates feedback control,* Bioinspiration & Biomimetics **12**, 036006 (2017).

[13] P. Zdunich, D. Bilyk, M. MacMaster, D. Loewen, J. DeLaurier, R. Kornbluh, T. Low, S. Stanford, and D. Holeman, *Development and testing of the mentor flapping-wing micro air vehicle,* Journal of Aircraft **44**, 1701 (2007).

[14] C. Richter and H. Lipson, *Untethered hovering flapping flight of a 3d-printed mechanical insect,* Artificial life **17**, 73 (2011).

[15] Q. V. Nguyen, W. L. Chan, and M. Debiasi, *Performance test of a hovering flapping wing micro air vehicle with double wing clap-and-fling mechanism,* in *International Micro Air Vehicle Conference and Competitions (IMAV)* (2015).

[16] Q.-V. Nguyen, W. L. Chan, and M. Debiasi, *Hybrid design and performance tests of a hovering insect-inspired flapping-wing micro aerial vehicle,* Journal of Bionic Engineering **13**, 235 (2016).

[17] C. De Wagter, S. Tijmons, B. D. W. Remes, and G. C. H. E. de Croon, *Autonomous flight of a 20-gram flapping wing MAV with a 4-gram onboard stereo vision system,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 4982–4987.

[18] G. C. H. E. De Croon, K. M. E. De Clercq, R. Ruijsink, B. Remes, and C. De Wagter, *Design, aerodynamics, and vision-based control of the delfly,* International Journal of Micro Air Vehicles **1**, 71 (2009).

[19] J. L. Verboom, S. Tijmons, C. De Wagter, B. D. W. Remes, R. Babuska, and G. C. H. E. de Croon, *Attitude and altitude estimation and control on board a flapping wing micro air vehicle,* in *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (IEEE, 2015) pp. 5846–5851.

[20] G. C. H. E. de Croon, M. Perçin, B. D. W. Remes, R. Ruijsink, and C. de Wagter, *The DelFly: Design, Aerodynamics, and Artificial Intelligence of a Flapping Wing Robot* (Springer Netherlands, 2016).

[21] H. Tokutake, S. Sunada, and Y. Ohtsuka, *Active control of flapping wings using wing deformation,* Transactions of the Japan Society for Aeronautical and Space Sciences **52**, 98 (2009).

[22] H. V. Phan and H. C. Park, *Generation of control moments in an insect-like tailless flapping-wing micro air vehicle by changing the stroke-plane angle,* Journal of Bionic Engineering **13**, 449 (2016).

[23] C. De Wagter, M. Karásek, G. de Croon, J.-M. M. G. Hattenberger, and H. de Plinval, *Quad-thopter: Tailless flapping wing robot with 4 pairs of wings,* (2017).

[24] S. B. Fuller, E. F. Helbling, P. Chirarattananon, and R. J. Wood, *Using a MEMS gyroscope to stabilize the attitude of a fly-sized hovering robot,* in *IMAV 2014: International Micro Air Vehicle Conference and Competition 2014, Delft, The Netherlands, August 12-15, 2014* (Delft University of Technology, 2014).

[25] E. F. Helbling, S. B. Fuller, and R. J. Wood, *Pitch and yaw control of a robotic insect using an onboard magnetometer,* in *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (IEEE, 2014) pp. 5516–5522.

[26] S. S. Baek, F. G. Bermudez, and R. Fearing, *Flight control for target seeking by 13 gram ornithopter,* in *IEEE/RSJ Int Conf on Intelligent Robots and Systems.* (2011).

[27] G. C. H. E. de Croon, M. A. Groen, C. de Wagter, B. D. W. Remes, R. Ruijsink, and B. W. van Oudheusden, *Design, Aerodynamics, and Autonomy of the DelFly,* Bioinspiration and Biomimetics **7** (2012).

[28] S. Sane, *Review: The aerodynamics of insect flight,* The journal of Experimental Biology **206**, 4191 (2003).

[29] J. E. Volder, *The cordic trigonometric computing technique,* Electronic Computers, IRE Transactions on , 330 (1959).

7 Discussion, Conclusions, Recommendations

At the start of this research, no small and lightweight MAV had ever been demonstrated that was able to perform all levels of autonomous flight. For the specific case of flapping wing MAVs, very little work had been done on the autonomous flight levels dealing with obstacle avoidance and navigation. For these reasons, the following research goal was formulated:

Research Goal

Develop autonomous flight capabilities for lightweight flapping wing Micro Air Vehicles.

To reach this goal, three research questions were posed. Section 7.1 discusses the approach taken and results obtained in addressing these questions. Section 7.2 draws final conclusions in light of the research goal and states the relevance of this research based on the main contributions. Section 7.3 proposes future work.

7.1. Discussion

The main results and conclusions from the corresponding chapters are discussed for each of the three research questions, followed by formulations of the final answers.

7.1.1. Using stereo vision

The first research questions was formulated as:

Research Question 1

How can stereo vision be used on flapping wing Micro Air Vehicles for autonomous flight tasks?

Stereo vision algorithms are meant to estimate distances to objects that are visible in stereo images. They produce disparity maps, which contain information about the 3D structure of the environment around the sensors. This information can

be used for all levels of autonomous flight. Previous studies showed that optical flow based algorithms, which are in essence a form of monocular vision, in combination with flapping wing MAVs and standard small cameras, can lead to performance issues when doing obstacle detection.

In Chapter 2, it is tested whether the performance of obstacle detection and avoidance improves when stereo vision is applied. A 5.2 gram stereo vision system is added to the DelFly flapping wing platform. It sends analogue video to a ground station that runs a stereo vision algorithm and produces heading control commands which are sent back to the vehicle. Results from this setup show that the influence of platform motion is significantly reduced because stereo cameras can provide images from different view points made at the same point in time. It is found to be crucial for this purpose that the stereo vision cameras run synchronous.

In flight, the stereo vision system yields distance estimates with a standard deviation of 20 cm at a distance of 350 cm. This result is obtained using an existing stereo vision algorithm that runs in realtime (25 FPS) on a desktop PC. Worse but adequate accuracy is still provided in situations where optical flow does not provide any information, even theoretically spoken. The disparity maps produced by the stereo vision algorithms are used as an input to test three different obstacle detection and avoidance algorithms. Two of these algorithms use purely reactive avoidance methods found in the literature that do not guarantee collision-free flight in all situations. Flight tests demonstrate that such situations indeed occur in the unadapted test flight environment. A new obstacle avoidance method is therefore proposed that is expected to provide collision-free avoidance maneuvers at any time, in any situation. Collision-free flights were recorded with durations up to 72 seconds. These tests show the great potential of using stereo vision for obstacle detection and avoidance, even though this system relies on off-board processing.

Due to the mass of the stereo vision system, which includes a separate 1 gram battery for enabling more stable video transmission, the maximum flight endurance of the DelFly Explorer reduces to 3 minutes. Furthermore, in this setup, the vehicle weighs 21 grams which is too heavy for hovering flight. The vehicle in this study includes a T-tail with an elevator and rudder. Flight tests reveal that using the rudder for heading control while flying in slow forward flight (i.e., with high pitch angles) creates oscillations around the body Z-axis. This is disadvantageous for the vision processing, and results in a poor turn rate control performance.

Bringing stereo vision on board of the vehicle drastically increases the required processing power, but also significantly reduces delays between sensor perception and actuator action, as it avoids any communication to and from the ground station. Limitations due to communication range will not play a role in such a setup, while all issues related to transmission noise on the images are solved. Furthermore, the vehicle can only be regarded as being fully autonomous if it does not rely on a ground station. Fully onboard integration of stereo vision on the DelFly flapping wing MAV is therefore demonstrated in Chapter 3. A custom-made stereo vision system of 4 grams is introduced which includes all functionality to allow onboard obstacle detection and avoidance. The camera system directly communicates with the autopilot board. The autopilot board is custom-made; it includes all functionality

required for autonomous flight and has a weight of 1 gram. The whole vehicle is powered by a single battery and has a total weight of 20 grams.

The design of the vehicle in this study, called 'DelFly Explorer', comprises a new tail design and additionally active aileron surfaces. The tail has no active rudder part, only passive vertical surfaces for damping oscillations around the body Z-axis. This configuration provides an improved heading control performance (higher rates, smaller overshoot) and reduced coupling effects. Adaptations to the brushless motor, in terms of number of windings per stator coil, result in an increase in power output. This enables the vehicle to fly at lower battery voltage levels, resulting in much longer flight times. Flight times up to 9 minutes have been recorded with this system.

A stereo vision algorithm is introduced called "LongSeq" which is suitable for running on the stereo vision system. It is a line-based stereo vision algorithm that is efficient enough to deal with the constrained processing power and available memory of the system. The line-based implementation performs a one-dimensional optimization over the image in order to produce dense disparity maps, even when the level of image texture is low. Dense disparity maps are useful for obstacle avoidance tasks as they allow to identify whether a texture-poor image region belongs to a nearby object or an object in the background. The LongSeq algorithm runs at ~11Hz allowing real time obstacle detection and decision-making by the vehicle. This line-based algorithm further allows for sub-sampling by skipping complete image lines, yielding a linear increase in processing speed at a negligible cost in accuracy. Decision making is based on a reactive control algorithm which identifies whether the vehicle is approaching an obstacle, and in which direction the vehicle should turn based on the location of the obstacle in the camera image. This approach has been tested in tests with sparse obstacle fields, where the obstacles are separated sufficiently to allow space for evasion. Furthermore, by combining flapping frequency control and barometric pressure feedback the vehicle is able to perform automatic take-off and closed-loop height control.

In Chapter 4, the avoidance method of Chapter 2 is combined with the stereo vision system of Chapter 3 on board of the DelFly Explorer. This study shows why the proposed method is suitable for this specific type of platform. In the first place, the method matches the nonholonomic constraints of the flapping wing MAV, and therefore allows it to better exploit its flight efficiency in slow forward flight. Second, the method does not conflict with important limitations of the cameras. Even though the field-of-view of the cameras is limited, the method guarantees free maneuver space without the need to combine camera observations in different directions. The method does not require to compute a trajectory in between objects, which makes the low resolution of the cameras less of an issue. It better suits more complex environments, such as dense obstacle fields and environments with walls. Furthermore, the method allows to use a different, sparse stereo vision algorithm. This algorithm is not intended to localize the size and position of obstacles within the image, but rather intends to robustly identify whether an obstacle is present in the image and at what distance it is found. The avoidance strategy also allows to efficiently combine these observations to obtain a more robust detection perfor-

mance. Motion blurring effects due to avoidance maneuvers are kept at a low level by preventing the need for aggressive maneuvers.

It is possible to extend the method to three dimensions by including vertical maneuvers. The vertical maneuvers form an additional avoidance maneuver without conflicting with the horizontal maneuvers. Using the current stereo vision system the height control performance is not sufficiently adequate for implementing this on the real system, however. The barometric pressure sensor does not provide the required performance. For height control purposes the observations from the stereo vision system are less reliable, as the texture of floors and ceilings is often low, and the distances at which these surfaces are typically observed (taking into account the small vertical field-of-view) are relatively large.

The complexity of the avoidance strategy is low in terms of processing and memory requirements, such that the majority of the processor capacity can be used by the stereo vision algorithm. This is beneficial for such an embedded system. The total obstacle detection loop runs at over 15 Hz on the system. Depending on the type of environment, flight times of 9 minutes without collisions can be achieved by this approach.

Based on the findings from Chapters 2 to 4 an answer to the first research question can be formulated. It is shown that stereo vision enables for robust obstacle detection and avoidance on board of a flapping wing MAV. It provides a solution to problems encountered when using an optical flow-only approach with the current state-of-the-art of small camera technology. An important issue with using cameras is that platform motions (vibrations, flapping and steering) affect the quality of the captured images and the location of image features within the image. Synchronized stereo vision cameras reduce these effects considerably. Platform motions still decrease the image quality compared to static observations, but such a stereo vision setup improves the obstacle detection performance while approaching these obstacles. Stereo vision as primary sensing input is found to be a crucial starting point to realize a high performance and robustness.

It was found that the control and behavior of the platform are strongly related to the characteristics of the stereo vision system and the platform. These relations are visualized in Figure 7.1. The obstacle detection performance of the stereo vision system degrades with higher maneuvering speeds of the vehicle **(a)**. The space observed by the cameras is limited, which prohibits certain maneuvers **(b)**. To obtain robust obstacle avoidance performance, the avoidance strategy therefore needed to be re-designed to cope with these restrictions. The avoidance strategy also affected the choice for the used stereo vision algorithm, as the strategy defines the specific requirements for the vision algorithm **(c)**. The chosen stereo vision algorithm allows the vehicle to maintain its forward speed during maneuvers **(d)**, as it runs fast enough to detect safe flight directions while performing a turn. This has a positive effect on the flight endurance of the vehicle **(e)**. The vehicle design was also changed to adapt the behavior of the vehicle. By means of the aileron design, which replaces the rudder, the vehicle dynamics allow for smoother heading control. This results in a better execution of the intended avoidance maneuvers **(f)** and reduces tail oscillations that affect the performance of the stereo vision system

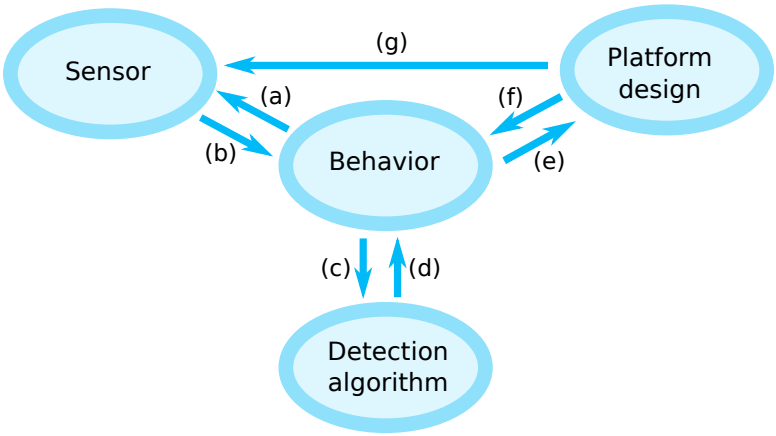

Figure 7.1: Diagram visualizing the relations between different system elements. These elements altogether are needed to realize autonomous flight capabilities. It was found in this thesis that the shortcomings of a certain element can be compensated by (a combination of) other elements.

(g). These findings clearly show that the combination of vehicle design, sensor selection, sensor processing algorithms, control behavior and also assumptions on the type of environment all have to be considered to optimize the performance of the obstacle avoidance task.

7.1.2. Learning from collisions

The second research questions was formulated as:

Research Question 2

How can a monocular robot learn by itself to see distances to obstacles by means of appearance?

In Chapter 5, a study is performed with a platform that is similar to the flapping wing MAV used in Chapters 3 and 4. Instead of a stereo vision system, the vehicle is equipped with a monocular vision system of 2 grams. To learn distances from monocular images, corresponding distance estimates need to be obtained via other sensors. These estimates are obtained when the vehicle has a (near-)collision with the environment. By assuming a constant flight speed, retrograde extrapolation also produces distance estimates for the images prior to the (near-)collision.

Flight tests where physical contact between the vehicle and the environment is detected via the accelerometers show that recovering from a real collision is difficult for the vehicle in its current shape. An additional infrared proximity-sensor is therefore added to the camera system that detects near-collisions and allows the vehicle to turn away before physical contact is made. Since the memory available on board of the camera system can only store a small number of images, an efficient image description algorithm is used to compress the image data. This is essential

for the learning process, which requires sufficient training data.

The image description algorithm used in this study is meant to describe the appearance of what is visible in the images. It is chosen specifically because it is known that the appearance of a certain scene changes over distance. In order to relate image descriptors with distance estimates, it was found that a standard k-NN approach is effective only when a large amount of training samples is stored, which is not feasible on the camera system used in this study. A clustering step is therefore included in the training process that maximizes the amount of stored training data while stimulating that the variation in image appearance types remains high.

Computer simulation results show that in confined spaces the learning performance after including the clustering step is only marginally reduced. The simulations also show that by using the learning data for estimating distances from new image frames allows the vehicle to prevent near-collisions from happening. Over time the frequency of such events indeed reduces. It is also observed that the avoidance behavior can be different if the vehicle is not allowed to use the training data for collision avoidance right from the start of a flight. By allowing an initial period of pure training, the learning performance increases faster, and the number of false-positives (avoiding too early), is reduced. Real flight tests indicate that the learning rate is lower compared to the computer simulation results, but similar trends are still observed. By storing training data over several flights, the vehicle is able to perform flights with successful obstacle avoidance in a confined room.

The findings from Chapter 5 confirm that the proposed implementation serves as an initial answer to the research question. Test results indicate that distance information can be learned and stored. At the same time it is shown that directly implementing a standard machine-learning approach, such as k-NN, is very challenging and not feasible on a small embedded system. The amount of training data that can be stored is extremely limited. For specific applications, for instance those where the MAV operates always in the same small space, the proposed method can form an effective solution for collision avoidance and localization. However, the general applicability of the proposed implementation remains to be shown. Since the current experiments show that learning performance is not perfect for the small space where the tests were performed, it is obvious that this performance will reduce when the flight space increases.

Replacing stereo vision by monocular vision, which is beneficial in terms of weight reduction, is neither a feasible nor a practicable approach to circumvent the use of optical flow for detecting and avoiding obstacles. However, the findings from this study give insight in the potential of the learning approach. As the hardware of the monocular vision system and the stereo vision system is identical, except for the number of cameras, the current method can be combined with the stereo vision system. In such a setup, the distance learning principle can complement the obstacle detection process. In situations where stereo vision fails to detect an obstacle, for example a transparent window, the learning method can recognize the same situation at a later moment in time. Since these are generally very specific and infrequent situations, the memory requirements for this method can be kept low. Besides, the additional proximity sensor is found to be a useful

additional sensor for obstacle detection in the first place. Because it will only serve as a last resort to avoid crashes abruptly (in a risky way), it is not suitable as a replacement for (stereo) vision, but rather as a complement.

7.1.3. Improving control authority

The third research question was formulated as:

Research Question 3

Can the performance of the obstacle avoidance task and other navigation tasks be improved by increasing the control authority of flapping wing Micro Air Vehicles?

In Chapter 6, an innovative control mechanism is proposed which is primarily intended to improve the heading control performance of the flapping wing MAV when in hover. At low flight speeds, the effect of the ailerons on the heading rate is very weak, making it difficult to maintain a preferred heading angle. Especially when obstacles are detected too late to perform a standard avoidance turn, the ability to hover and turn around in approximately the same position would be beneficial as it provides an alternative and perhaps very versatile and robust escape maneuver. Static tests on a force/moment balance show that considerable control moments for heading control can be produced by the proposed mechanism. Flight tests show that indeed high heading rates can be achieved and that the vehicle is able to reverse its flight direction while requiring only a small turn space. The proposed mechanism allows to control the pitch and roll angles as well. Hence, although at the cost of using an extra servo, and heavier/stronger servos, the attitude of the vehicle can be controlled fully without the need of a tail. Results indicate that the effectiveness of the proposed control mechanism is substantially smaller for roll than for pitch and yaw.

The vehicle design as presented in Chapter 6 is likely to improve the platform's obstacle avoidance performance considerably. The proposed control system extends the flight envelope with the hovering capability. This allows the system to perform turns on the spot, providing an additional obstacle avoidance maneuver. Results from Chapter 4 also show that by being able to perform smaller turns, the covered area during exploration flight increases. Furthermore, the improved attitude control performance enables the vehicle to better execute the intended flight path. Wind disturbances, such as those generated by climate control systems, will have a less dramatic effect on a vehicle with a low wing loading. The flight envelope is further extended with sideways flying. Such a flight capability can be used in other navigation tasks, such as following the center line of a narrow corridor or flying through a door. The vehicle design in this study is heavier than the design from Chapters 3 and 4. However, taking into account potential design improvements, the extra weight of the vehicle including stereo vision camera will still be less than 0.5 gram.

7.2. Final Conclusions

This dissertation shows effective methods that have been designed to perform the first three levels of autonomous flight: attitude control, height control and collision avoidance.

Attitude estimation and control are feasible on a flapping wing MAV using the proposed redesign of the vehicle by adding a new control mechanism. This mechanism improves actuator effectiveness on all three axes and extends the flight envelope with the abilities to perform stable hovering flight and sideways flight.

For height control, a barometric sensor can provide a height estimate with reasonable performance. In contrast, the measurements from the stereo vision system as used in this dissertation do not provide improved height estimates in most general situations.

Stereo vision on a lightweight Flapping Wing MAV is shown to be a feasible and also very effective solution to performing obstacle avoidance. It definitely outperforms optical flow based solutions. Obstacle detection and avoidance based on monocular vision through learning is conceptually possible, but several limitations and issues related to this approach make this solution currently less effective and less robust compared to stereo vision.

For the fourth level of autonomous flight, navigation, valuable insights have been obtained. Simultaneous Localization And Mapping (SLAM) approaches do not form a good basis on a lightweight flapping wing platform; not only because of severe restrictions to the available amount of onboard processing power and memory, but also because of the combination of the extremely agile platform dynamics and poor sensor performance of current small and low-weight camera technology. Results indicate that the principle of learning the appearance of the environment is useful for the task of recognizing earlier visited places. This is an important ability when the navigation goal is to explore as much different places as possible within a certain (often limited) flight time.

Furthermore, the extended flight capabilities allow for more and also very different vehicle maneuvers. These are not only valuable to improve the obstacle avoidance performance, but also allow to perform specific tasks in narrow spaces, such as flying through narrow corridors and opened doors and windows. It dramatically increases the versatility of the vehicle.

In the process of finding answers to the research questions in this dissertation, the strong relationships between sensing, processing, the vehicle design, its behavior and the environments have become clear. In the development of the algorithms and mechanisms needed to reach the research goal, it was often necessary to take these relationships into account in order to enable the system to execute certain tasks. This clearly indicates that the concept of Embodied Intelligence, which states that intelligent behavior emerges from the interplay between brain, body and world, is important when developing an autonomous system. While such relationships are often found by reasoning, some will only be found by bringing these elements together in a real system. Hence, reaching an efficient approach to a challenging task in robotics benefits from an experimental approach. It not only allows the developer to take such relationships into account, but also enables the developer

to take advantage of them in order to reach more efficient solutions.

Such an approach is often not applied in the field of autonomous MAV research. The common approach where SLAM is applied to a multicopter is an example where the sensing and processing parts are isolated and optimized for performance in terms of precision. Because of this precision such vehicles are able to perform advanced navigation tasks, but it is not analysed whether the obtained solution is efficient in terms of overall performance. Especially when it comes to the development of lightweight and extremely small MAVs, efficiency and performance are extremely important.

7.3. Future Work

The follow-up of the work performed in this dissertation can be split into three main parts. First, the integration of the stereo vision system developed in the first chapters with the vehicle equipped with the control mechanism proposed in Chapter 6. Second, the development of smaller versions of the flapping wing MAV with the same autonomous capabilities. Third, extending the current autonomous capabilities with more advanced navigation capabilities to reach full autonomy on all levels.

In this dissertation, several algorithms and mechanisms have been developed. Furthermore new electronics and sensors were introduced. The first next step would be to combine the stereo vision system with the control mechanism, thereby integrating all autonomous capabilities and flight capabilities into one system. This may involve design iterations to limit the increase in total weight of the system. It is estimated that this increase will be limited to 0.5 gram, but this needs to be confirmed. Because the effectiveness of the roll control mechanism is insufficient to cope with small disturbances, it needs to be investigated if its effectiveness can be improved. Other roll control mechanisms might also be explored that can be combined with the current mechanism for pitch and yaw.

A subsequent step would be to develop a smaller and lighter version of the same system. An important aspect of this development will be on a further miniaturization of the payload of the vehicle: the autopilot board and the stereo vision system. Both are currently among the smallest of their kind. This can be realized by redesigning them (using smaller electronic parts, which also depends on newer generations of the used electronics), by simplifying their design by removing components, and perhaps by combining both systems in one design. By realizing a smaller payload weight, a smaller version of the current flapping wing MAV becomes viable that has the same performance in terms of flight time. It is expected that the concept of using stereo vision for obstacle avoidance is scalable. A smaller stereo vision system will obviously have a smaller baseline. As a result the depth resolution of such a system will become smaller, but this scales with the size and, more importantly, the lower flight speed of smaller vehicles. The concept of the developed avoidance strategy is expected to remain relevant, as even very small vehicles will need sufficient obstacle-free space to perform avoidance maneuvers. It was found that the limitations of sensors influence the types of avoidance maneuvers that can be performed safely. Especially sensors that are even smaller and

more compact than the current state-of-the-art are expected to have more severe limitations. A control strategy that takes all these limitations into account, as is developed as part of this dissertation, would then be inevitable.

The third step that can be taken as a continuation of this work, is to expand the current autonomous flight capabilities to the fourth level: navigation. Since navigation is a broad term, navigation tasks in the context of this research are focused on that would require a small number of additional sensors and design changes. In the first place, the concept of learning from the appearance of the environment, as was mentioned previously in Section 7.1.2, can be used to recognize places where obstacle detection is not reliable, or to recognize earlier flown flight patterns. These features will make the vehicle more reliable in terms if obstacle avoidance, and enables it to more effectively reach unexplored places. In the second place, recently developed small Time-of-Flight ranging sensors are expected to enable various autonomous flight capabilities. These small sensors, weighing less than 0.1 gram each, are specifically useful for indoor navigation, as they deliver range estimates at ranges of 1 to 2 meters, and only if there is no direct sunlight.

On the flapping wing MAV developed as part of this dissertation, these sensors can be used for a range of tasks: to improve height control performance, to measure the distance to objects on the side outside the field of view of the forward facing stereo vision system (useful for wall following, door detection, narrow corridor traversal) and to detect obstacles that cannot be detected by cameras, such as texture-poor objects that have a similar color as the background and transparent windows. These sensors thus have the potential to make current autonomous flight capabilities more reliable. Ultimately, these sensors enable the platform to perform the most important navigation capabilities that are required to perform autonomous exploration flight in complex environments.

Acknowledgements

This dissertation is the result of a process of more than four and a half years. During this period it has been a big part of my life. Now, at the end of this period, I would like to thank all the people that contributed to that part of my life, and also those people that were important in other aspects of this life.

I would like to express my deepest gratitude to my supervisor and co-promotor, Dr. Guido de Croon. As a researcher he was always open for discussing new ideas from the both of us, and I have always appreciated his enthusiasm for making them converge into new research questions together. As a supervisor he played an important role in helping me making decisions in the research process. Not to limit creativity, but to form realistic and manageable work plans. I have always appreciated his effort in providing me with constructive feedback on basically everything I wrote during these years. Furthermore, he played the role of mentor and coach, by stimulating and helping me to improve my research skills, by sharing personal insights and experiences, and by motivating me during periods of adversity. He not only taught me how to perform relevant research, but also how to enjoy it.

I sincerely thank Prof. dr. ir. Max Mulder, my promotor. He not only allowed me to pursue my PhD, but insisted on starting it and played an important role in making it a success. I admire your efficiency in giving me feedback on my manuscripts, your experienced academic view that helped me to improve my work, and your prudent and persistent attitude in dealing with criticism from reviewers. In your own way you stimulated my confidence in my own work and helped me continuing this long process to the point where we are now.

I want to thank the early MAVLab team members: Bart Remes, Christophe de Wagter, Erik van der Horst, Rick Ruijsink and of course Guido de Croon. Their enthusiasm for drones, and for the DelFly project in particular, made me enjoy working with them and motivated me to do PhD research in this field. It has been a fascinating experience to do research on drones and witness how this new technology found its way into useful applications at an incredible rate during this period. Working with this team of skilled and experienced experts on various cutting-edge aspects of this technology has been a very valuable learning process. Their expertise in developing custom-made electronics enabled me to perform my research on a really unique platform.

That being said, I would like to express my sincere appreciation to everyone I have met and worked with in the MAVLab. No matter they were postdoc, PhD student, graduate student or intern, it often felt like they all were colleagues, sharing the same enthusiasm and drive to perform research on drones and to come up with (crazy) new ideas for their future applications. There was always a good atmosphere in the lab, and especially during the conferences and competitions we attended together. I would like to thank as many of you as I can remember right

now: Kevin van Hecke, Andries Koopmans, Hann Woei Ho, Freek van Tienen, Ewoud Smeur, Lodewijk Sikkel, Jochem Verboom, Kimberley McGuire, Kirk Scheper, Matěj Karásek, Kevin Lamers, Roland Meertens, Elisabeth van der Sman, Bart Slinger, Mario Coppola, Clément Roblot, Maxime Collette, Sabrine Ahmed Iqbal, Volker Strobel, Torbjørn Cunis, Deon Blaauw, Tom van Dijk, Titus Braber, Michaël Ozo, Shuo Li, Wilco Vlenterie, Wilco Schoneveld, Diana Olejnik, Karl Martin Kajak and Aadithya Sujit.

Furthermore, I would like to thank everyone else from the Control & Simulation department. Our 'labmates', the colleagues from the research aircraft, who (almost) never complained about our drone-activities in the lab. All academic staff, supporting staff, and, of course, all current PhD students and those who can call themselves doctor already. These are really too many people to mention, but I would like to thank you all for your positive attitudes, interesting conversations, and also for the way how you are there for each other.

Apart from everyone I met in Delft, there are so many others that have been important to me during the last few years. These people were important as they helped me to get away from work when necessary. First of all, my family. My parents, brothers and sister, who are always there for me, who showed interest in my work, but also stimulated me not to get lost in my work. My family in law, who, in contrast to the clichés, always provides a warm and comfortable atmosphere, who are concerned with what everyone is doing, and gave me support in many different ways. Furthermore, all friends who were there form the start, and also those I met during the last few years. Thank you for your interest in my work, but also for not asking about my progress each and every time we met. You all really helped me to stay down to earth, thereby making me feel very comfortable. I thank all family and friends for sharing everything that is really important in life.

Finally, my greatest thanks go to my love, Jessica. You played the most important role in my life during the last four and a half years. We got married during this period, and started a new phase in our lives together. You always supported me in my PhD work (also by making me aware of my own commitments regarding planning), and, more important, simply made my live colorful. You are the main reason why this period in my life has been a very happy time. Another new phase in our lives is starting now. Not so much because I finished this dissertation, but mainly because of our daughter Julia, who was born two weeks ago. I look forward to the years to come with the three of us, and I hope we can give her as much love as we have for each other.

Curriculum Vitæ

Sjoerd Tijmons

08-10-1987 Born in Rotterdam, The Netherlands.

Education

2001–2006 Secondary School
Wartburg College, Rotterdam, The Netherlands

2006–2009 BSc in Aerospace Engineering
Delft University of Technology, The Netherlands

2009–2012 MSc in Aerosapce Engineering
Delft University of Technology, The Netherlands
Thesis: Stereo Vision for Flapping Wing MAVs

2013-2017 PhD in Aerospace Engineering
Delft University of Technology, The Netherlands
Thesis: Autonomous Flight of Flapping Wing Micro Air Vehicles
Promotor: prof. dr. ir. M. Mulder

Awards

2009 Distinguished bachelor student

2013 Best graduate of TU Delft in Aerospace Engineering

List of Publications

10. **S. Tijmons**, M. Karásek, G.C.H.E. de Croon, *Attitude control system for a lightweight flapping wing MAV*, In: Bioinspiration & Biomimetics (SUBMITTED).

9. **S. Tijmons**, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, M. Mulder, *Obstacle Avoidance Strategy using Onboard Stereo Vision on a Flapping Wing MAV*, In: IEEE Transactions on Robotics, (2017).

8. K. Lamers, **S. Tijmons**, C. De Wagter, G.C.H.E. de Croon, *Self-supervised monocular distance learning on a lightweight micro air vehicle*, In: International Conference on Intelligent Robots and Systems (IROS), (2016).

7. K.Y.W. Scheper, **S. Tijmons**, C.C. de Visser, G.C.H.E. de Croon, *Behavior trees for evolutionary robotics*, In: Artificial Life, (2016).

6. J.L Verboom, **S. Tijmons**, C. De Wagter, B.D.W. Remes, R. Babuska, G.C.H.E. de Croon, *Attitude and altitude estimation and control on board a Flapping Wing Micro Air Vehicle*, In: IEEE International Conference on Robotics and Automation (ICRA), (2015).

5. J.A. Koopmans, **S. Tijmons**, C. De Wagter, G.C.H.E. de Croon, *Passively stable flapping flight from hover to fast forward through shift in wing position*, In: International Journal of Micro Air Vehicles, (2015).

4. C. De Wagter, **S. Tijmons**, B.D.W. Remes, G.C.H.E. de Croon, *Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system*, In: IEEE International Conference on Robotics and Automation (ICRA), (2014).

3. **S. Tijmons**, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, H.M. Ruijsink, E. van Kampen, Q.P. Chu, *Off-board processing of Stereo Vision images for Obstacle Avoidance on a flapping Wing MAV*, Proceedings of the PEGASUS - AIAA Student Conference, (2014).

2. **S. Tijmons**, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, H.M. Ruijsink, E. van Kampen, Q.P. Chu, *Stereo Vision Based Obstacle Avoidance on Flapping Wing MAV's*, Advances in Aerospace Guidance, Navigation and Control, (2013).

1. **S. Tijmons**, G.C.H.E. de Croon, B.D.W. Remes, C. De Wagter, H.M. Ruijsink, E. van Kampen, Q.P. Chu, *Stereo Vision Based Obstacle Avoidance on Flapping Wing MAV's*, Proceedings of the EURO GNC, (2013).

Remarks on publications that are selected as chapters

Five publications were selected as chapters in this dissertation: **2**,**4**,**8**,**9**,**10**. These have been selected because of their direct link with the main research question. These publications also represent the work in which the author was the most involved in performing the actual research and conducting experiments. In those publications where he is the first author (**2**,**9**,**10**), and also in publication **4**, the author performed the main work, both on the hardware and software parts, except for the development of custom-made electronics. For publication **8**, much work, both on the hardware and software side, has been performed by Kevin Lamers, who was under supervision of the author. The author was directly involved in the realization of the publication, both in working out the research question and also in conducting the experiments.