# Adaptive Visual Servoing Control for Quadrotors: A Bio-inspired Strategy Using Active Vision

Sander Hazelaar

TU Delft

# Adaptive Visual Servoing Control for Quadrotors: A Bio-inspired Strategy Using Active Vision

Thesis report

by

Sander Hazelaar

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on Februari 27, 2024 at 10:00

*Thesis committee*:

| | |
|---|---|
| Chair: | Dr. Ir. C. de Wagter |
| Supervisors: | Prof. Dr. G.C.H.E. de Croon |
| | M. Yedutenko |
| External examiners: | Dr. Ir. B.W. van Oudheusden |
| | Prof. Dr. F.T. Muijres |
| Place: | Faculty of Aerospace Engineering, Delft |
| Project Duration: | September, 2022 - February, 2024 |
| Student number: | 4552784 |

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

Faculty of Aerospace Engineering · Delft University of Technology

# Preface

More than a year ago, I started working on this thesis project, which combined my engineering background with a fascination for biology. The efficiency and precision in the way animals and humans control their bodies have always intrigued me, leading me to enroll in the Bio-inspired engineering course in my master, which fueled my desire to continue in this research field. For my thesis subject I contacted Guido and he introduced the very unique idea to try a new vision-based landing strategy on a drone that has been observed in bumblebees. This project has allowed me to get practical experience in writing real-time software for a small flying robot. Initially it was very challenging, but after a few months of experience I got the drone to fly with my own developed algorithms.

Alongside this master thesis I have been trying to pursue my dream to one day row at the Olympics. The combination of sports and studying has often not been easy and required me to improve my planning skills. However, reflecting on the past year, I am proud of what I have achieved both academically and athletically.

Recognizing the support that surrounded me during this project, I want to take the opportunity to thank the people who played crucial roles. Firstly, I want to thank Guido for his expert guidance in my research. Matthew, your motivational speeches, valuable academic insights and overall positive energy kept me going. Secondly, I want to thank my family for their constant support. Thirdly, I would like to thank my fellow rowers and coaches at Proteus for the good times both on and off the water. Lastly, I want to thank Hidde, Tim and Hajo for joining me on the thesis journey.

Sander Hazelaar
*Delft, Januari 2024*

# Contents

# Nomenclature

**List of Abbreviations**

| | |
|---|---|
| FOE | Focus of Expansion |
| GPS | General Positioning System |
| IBVS | Image-Based Visual Servoing |
| IMU | Inertial Measurement Unit |
| INDI | Incremental Non-linear Dynamic Inversion |
| LiDAR | Light Detection and Ranging |
| MAV | Micro Air Vehicle |
| NDI | Non-linear Dynamic Inversion |
| PBVS | Position-Based Visual Servoing |

**List of Symbols**

| | |
|---|---|
| $\mu$ | reference acceleration |
| $\omega_c$ | Cut-off frequency |
| $\phi$ | Roll angle |
| $\psi$ | Yaw angle |
| $\tau$ | Time-to-contact |
| $\theta$ | Pitch angle |
| $\vartheta$ | Optical flow divergence |
| $\xi$ | Position in inertial reference frame |
| $C$ | Center of landing target |
| $c_e$ | Gain scheduling factor |
| $c_{lp}$ | Low-pass filter scaling factor |
| $D$ | Optical flow divergence |
| $d$ | Euclidean distance to landing target |
| $F$ | Aerodynamic force |
| $f$ | Focal length |
| $g$ | Gravitational acceleration |
| $h$ | Height above landing surface |
| $k_d$ | Derivative gain |
| $k_i$ | Integral gain |
| $k_p$ | Proportional gain |
| $m$ | mass |
| $P$ | Image feature length |
| $p$ | Roll rate |
| $Q$ | Number of pixels in the image plane occupied by the landing target |
| $q$ | Pitch rate |
| $r$ | Yaw rate |
| $T$ | Total thrust force |
| $v$ | Time derivative of Euclidean distance to landing target |
| $w$ | Wind velocity |
| $Z$ | Image feature depth |

# List of Figures

# 1 Introduction

Micro air vehicles (MAVs) have been gaining popularity over the past two decades, owing to their versatility and wide range of applications. They have found uses in diverse areas such as agriculture, maintenance inspections, search and rescue, and aerial photography. The list of applications is expected to grow with improvements in MAVs' flight capabilities, autonomy and safety.

Traditional autonomous navigation methods rely on sensors such as GPS, LiDAR, or SONAR to estimate the position and orientation of the vehicle, define a waypoint in inertial space, and plan a path to the destination. While LiDAR and SONAR are effective in some situations, they are heavy and require significant computing power and energy. Moreover, GPS signals may not always be available. This dramatically reduces the performance and use cases for a flying robot, as a heavy drone can often be unsafe to operate close to humans. Therefore, reducing weight is one of the most productive ways to improve safety.

Some research is devoted to this, with good results, such as the Delfly [1]. Only low-power and lightweight sensors are available on these tiny drones to reach a high level of autonomy. For this, a monocular camera is suitable, as it passively obtains rich information from the environment. When using a monocular camera, the optic flow is a valuable source of information. This occurs when the observer moves relative to the environment, which results in a moving pattern of light hitting the retina. An extensive body of research has been performed on this phenomenon to use it for landing an MAV, but its use is still not widespread. The main reason is that a monocular camera can only measure angles to objects, but not their distance. Because of this, optic flow can only give information about the distance relative to the velocity, making landing with optical flow a non-linear control problem. In this problem, the controller depends on the distance to the target [2]. Therefore, the key challenge is to approximate the distance during the approach.

The solution might be found in nature, as flying insects can land accurately on flowers or onto their nest in a wide range of weather and lighting conditions while predominantly relying on vision and with minimal computational capabilities. In experiments with flying insects, it has been found that optical flow plays a crucial role in many flying tasks. Landing insects were thought to keep the optical flow expansion constant to make a smooth approach, but a recent study on bumblebees shows that the expansion is only constant for short periods and is step-wise increased when closing in on the target [3].

When landing, the insects not only have to reduce their speed while nearing the target but also need to steer their bodies to maintain the proper orientation. In robotics, this 3-dimensional visual control problem is called visual servoing. The term landing is often used to describe a simplified version of this problem where the only control parameter is the approach speed.

This thesis aims to improve existing visual servoing methods and our understanding of the control strategies of flying insects by implementing the adaptive landing behavior found in bumblebees on a quadrotor MAV. The bumblebee experiment from [3] will be mimicked to test whether this can lead to a practical control strategy.

## 1.1. Research Objective

The goal of this master thesis can be formulated with the following research objective:

**Research Objective**

Developing a bio-inspired visual servoing strategy, based on the behavior found in bumblebees, to perform landings with a quadrotor MAV, using only a monocular camera and an inertial measurement unit

## 1.2. Structure of the Report

This report is divided into three parts. First, in Part I the scientific article can be found, which shows the main contributions of the thesis. It starts with an introduction to the problem, signifying the relevance of the work and providing a summary of the contributions. Following this, the algorithms developed for the quadrotor MAV are described. Subsequently, the results of the real-world flight tests are presented. The article concludes with a brief summary of the results, and recommendations for further research are provided.

In Part II, the literature review is presented that was performed at the start of the thesis. Here, the most relevant research from two distinct research fields is explored. It starts with the visual navigation strategies of animals in Chapter 3. This is followed by the vision-based control methods in robotics in Chapter 4. Subsequently, Chapter 5 discusses the research bridging both fields. After this, the most important findings are synthesized in Chapter 6. The last section shows the conclusions of the literature review.

Last, Part III offers additional results to the scientific article and provides closing remarks. Chapter 8 analyzes the benefits and shortcomings of time-to-contact based landing strategies. Following this Chapter 9 presents an alternative method for the controller adaptation and its limitations for landing with visual servoing. The thesis is concluded in Chapter 10, summarizing the findings and offering recommendations for future research.

# Part I

## Scientific Article

# Adaptive Visual Servoing Control for Quadrotors: A Bio-inspired Strategy Using Active Vision

S.T. Hazelaar*, M Yedutenko, and G.C.H.E. de Croon
Aerospace Engineering Faculty, Delft University of Technology, Anthony Fokkerweg, Delft

## Abstract

New insights into the landing behavior of bumblebees show an adaptive strategy where the optical flow expansion of the landing target is step-wise regulated. In this article, the potential benefits of this approach are studied by replicating the landing experiment with a quadrotor. To this end, an open-loop switching method is developed, enabling fast steps in divergence. An adaptive control law is used to deal with non-linear system dynamics, where the control gain is scheduled based on the control effectiveness of the actuator inputs during the steps. It is demonstrated that the quadrotor can reliably land on the target from varying initial positions, and the switching strategy shows a slight reduction in landing time compared to a constant divergence strategy with the same average divergence over distance. This strategy also reduces the maximum velocity during the landing.

## 1 Introduction

There has been a growing interest in small flying robots in the last two decades due to their exceptional maneuverability. This allows them to reach locations inaccessible to humans. The potential applications of these robots span a diverse range, including agriculture, photography, maintenance inspection, and military operations. Despite their versatility, the operational use of these robots is still limited because they are often unsafe to fly close to humans or within indoor environments.

To address these limitations, the key objectives involve improving their autonomous flight performance and minimizing their weight. However, achieving complete autonomy in micro air vehicles (MAVs) is challenging due to constraints in payload capacity and computational resources [1]. Consequently, current research predominantly focuses on developing efficient algorithms tailored for various navigation tasks. A crucial maneuver to master for all flying robots is precise landing or hovering near objects. Possible scenarios are navigating to a docking station [2], inspecting windmills [3], or even pollinating flowers [4].

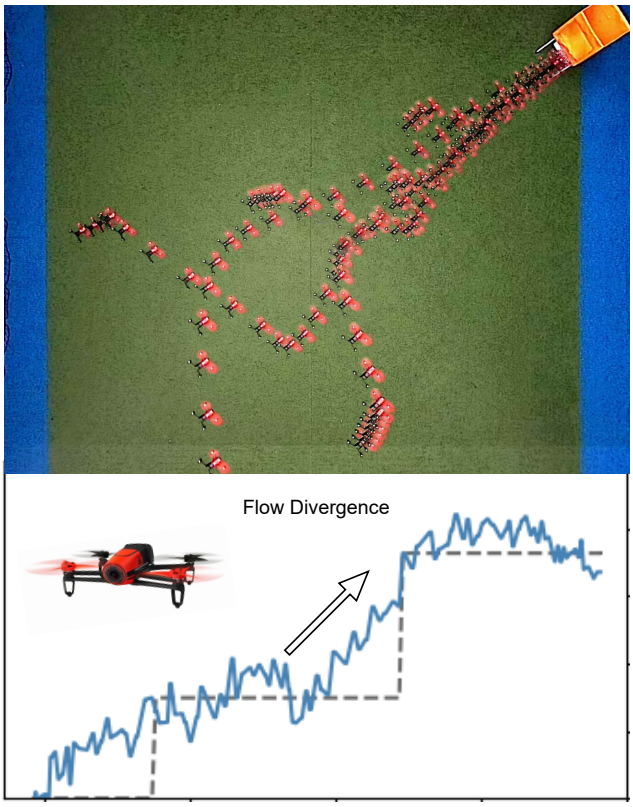


Figure 1: Quadrotor flight tests starting from varying initial positions using our visual servoing algorithm inspired by new landing behavior observed in bees [5]. The flow divergence set-point is step-wise increased when approaching the artificial hive, enabling faster landings. The step input response of the divergence signal is achieved through an open-loop control input to the motors of the quadrotor. All methods are tested on the Parrot Bebop robotic platform using the Paparazzi autopilot software. The computer vision and control software modules run on-board.

To execute these tasks, flying robots are often equipped with various sensors designed to perceive their environment. While GPS is a common choice, its availability is limited in proximity to buildings or within indoor environments. Moreover, GPS fails to measure the position relative to the surroundings directly. Ranging sensors, like LiDAR, can address this, but their weight and high computational demands pose challenges for MAVs [6]. In contrast, cameras offer a

lightweight alternative for passively acquiring rich information from the environment [7].

The process of regulating the position and orientation of a robot through feedback from a camera is known as visual servoing [8]. Utilizing camera images, the robot's position relative to surrounding objects in inertial space can be reconstructed and used as feedback to the system, termed position-based visual servoing. A more favored approach involves conducting all computations in the image domain, called image-based visual servoing (IBVS). IBVS is more robust as it is less susceptible to camera calibration errors and eliminates the need for additional sensors providing absolute values of distance or velocity.

When landing with IBVS, two properties of the visual elements in the image are important. The position of the elements in the image plane and their velocity, also called optical flow. Optical flow occurs when moving relative to the surroundings [9, 10]. When approaching an object, the optical flow indicates an expansion of the visual elements; its magnitude is the divergence. This is a measure of the ratio between the robots' velocity and distance relative to the approached object. Maintaining a constant divergence ensures a smooth landing, where the relative distance and velocity to the landing platform are exponentially reduced to zero. This seems to be an elegant strategy, as neither velocity nor distance has to be determined.

However, this also drives one of the critical challenges of IBVS for landing. The dynamics between the system control inputs and the visual output are highly non-linear and change with the distance to the surroundings [11]. While the ratio of velocity and distance can be measured from vision, the absolute distance remains unknown. At a large distance, control inputs generate minimal optical flow. Yet, as the robot nears an object, optical flow increases rapidly, making it challenging to maintain system stability.

A common solution is to adapt the controller during the landing to deal with the non-linear dynamics. In previous work, attempts have been made to apply a Lyapunov stability analysis to design a controller that enforces stability throughout the landing [12, 13, 14]. However, these analyses often neglect the real-world factors of the actuator- and vision delays, meaning that system stability in a real-world environment is not guaranteed in proximity to the landing target.

An alternative strategy involves estimating the distance based on the system dynamics [15]. This technique demonstrated with a camera sliding on rails, yields accurate distance estimates by leveraging the relationship between the observer's acceleration and the time derivative of divergence. In [16], the same relation is used to develop an adaptive incremental non-linear dynamic inversion (INDI) controller that can land a quadrotor. INDI is often called a sensor-based control method, as it relies on the measured inputs and outputs of the system to adapt the controller. Therefore, accurate sensor readings are crucial for successful implementation.

Another method for distance estimation involves the stability of the divergence control loop. It has been found that the distance at which instability arises is proportional to the chosen control gain [17]. It is demonstrated that this can be used to change the control gain or to trigger a landing. A drawback of this approach is that the system must reach instability to compute the distance.

In all the examples mentioned above, the landings are performed on large flat surfaces, which is a simplification over most real-world landings, where the precise landing location is critical. Landing on a small target introduces two complicating factors. First, the system has to control all degrees of freedom accurately. Second, the divergence of a small target is more challenging to measure, making the reliability of methods from prior research questionable.

Solutions to this problem might be found in nature, as we see animals gracefully landing on small surfaces everywhere around us. Examples are a pigeon landing on a small tree branch, a bumblebee landing on a flower, or a gannet diving onto its prey. During these maneuvers, optical flow is the most prominent source of information [18]. One of the most studied optical flow landings is that of bees, as they can be trained to fly specific routes. Remarkably, a bumblebee can perform up to 1000 landings on flowers per hour [19], indicating the need for an efficient and robust strategy.

A recent study looked at the trajectories of individual bumblebees and honeybees and found an adaptive strategy where the divergence is step-wise regulated during the landing [20, 5]. It was observed that the bees often increased divergence when approaching their landing target, potentially increasing the landing speed.

To check this hypothesis and to see how the discrete steps in divergence can improve MAV landings, we replicate the experiment from [5] on a quadrotor MAV. Figure 1 shows multiple runs from flight tests. The goal is to perform a horizontal landing in the orange cube (hive) from varying initial positions using only a single camera and the on-board inertial measurement unit (IMU).

To this end, the following contributions are made in this article. (1) A novel landing strategy is proposed, using visual servoing inspired by the landings of bees. By step-wise increasing the divergence set-points, slightly faster landings can be achieved while reducing the required maximum velocity. It uses (2) a non-linear controller, in which the control gain is inversely proportional to the control effectiveness during a switch in the divergence set-point. To improve the divergence measurements, (3) an adaptive low-pass filter that varies with the retinal size of the hive is introduced. All software components are validated on a real robotic platform with flight tests. The visual servoing controller was able to accurately position the quadrotor close to the artificial hive from varying initial positions.

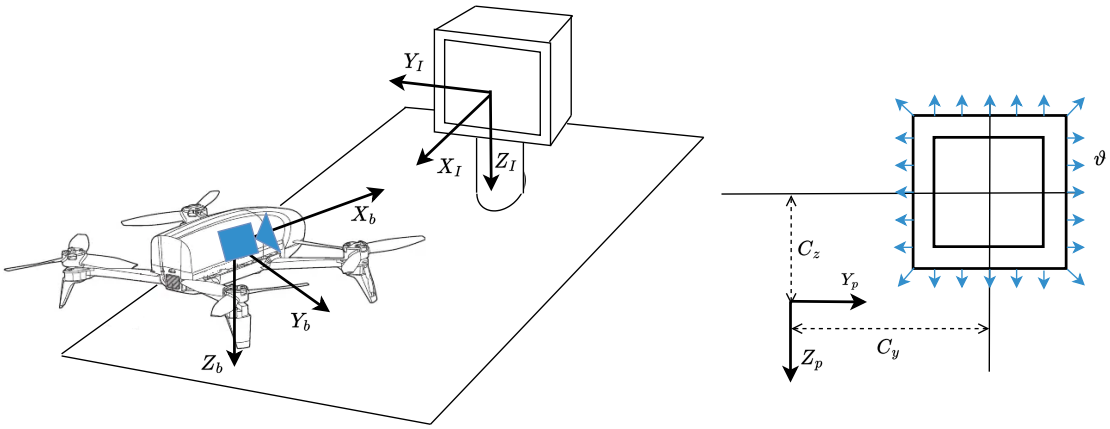


Figure 2: MAV body, inertial and image reference frames. The origin of the inertial reference frame is fixed to the center of the artificial hive.

## 2 Methodology

This section describes the components of the visual servoing autopilot. First, the quadrotor dynamics and the visual servoing control law are formulated. Second, the method is shown to estimate the optical flow divergence from the scale of objects in the camera images. This is followed by the proposed switching strategy. Lastly, we show the gain scheduling of the visual servoing controller.

### *2.1 Visual Servoing Controller*

Consider the situation shown in Figure 2. The position of the quadrotor $\xi$ is given in the inertial reference frame $(X_I, Y_I, Z_I)$, which has its origin fixed to the center of the artificial hive. The attitude $(\phi, \theta, \psi)$ of the quadrotor defines the rotation of the body frame $(X_b, Y_b, Z_b)$ with respect to the inertial frame.

The quadrotor uses a single camera pointing forward in the $X_b$ direction, which is assumed to be located at the center of gravity of the MAV. Figure 2 shows the image plane with the coordinates of the center of the hive defined as $C_y$ and $C_z$ and the optical flow divergence $\vartheta$ that occurs when approaching the hive, given by:

$$\vartheta = -\frac{v}{d} \tag{1}$$

With $d$, the Euclidean distance to the hive and its time derivative $v$. Note that $v$ is negative during the landing. Therefore, the minus sign is added to make the divergence positive when approaching the hive.

Now that the reference frames are defined, we can look at the quadrotor's control system. A quadrotor is an underactuated system, which can be seen as two connected subsystems. We can design a control law with an inner- and outer-loop. The outer-loop controls the position $\xi$ of the quadrotor by rotating the thrust vector. The inner loop handles the attitude through the angular velocity of the four propellers. For the outer loop controller, we can look at the position dynamics of a quadrotor, where we keep the yaw angle $\psi$ fixed:

$$\ddot{\boldsymbol{\xi}} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} \cos\phi \sin\theta \\ -\sin\phi \\ \cos\phi\cos\theta \end{bmatrix} T + \frac{1}{m}\mathbf{F}(\dot{\xi}, w) \tag{2}$$

Where $g$ is the gravitational acceleration, $m$ is the mass, $T$ is the total thrust force, and $F$ is the aerodynamic force, which is a function of velocity and wind $w$. Then, the following control law can be used to find the desired values $\boldsymbol{\mu}$ for $\ddot{\boldsymbol{\xi}}$ with feedback from the camera:

$$\mu_x = k_{p_x} * (\vartheta^* - \vartheta) + k_{i_x} * \int (\vartheta^* - \vartheta) \tag{3}$$

$$\mu_y = k_{p_y} * C_y + k_{d_y} * \frac{d}{dt}(C_y) + k_{i_y} * \int (C_y) \tag{4}$$

$$\mu_z = k_{p_z} * C_z + k_{d_z} * \frac{d}{dt}(C_z) + k_{i_z} * \int (C_z) + g \tag{5}$$

Where $\boldsymbol{k_p}$, $\boldsymbol{k_i}$, $\boldsymbol{k_d}$ are the proportional, integral and derivative gains respectively. $\vartheta^*$ is the divergence set-point, and the reference values for the centroid are set to 0. In this way, the centroid of the hive in terms of $C_y$ and $C_z$ is used to center the quadrotor on the hive and the divergence $\vartheta$ to regulate the forward acceleration.

Subsequently, the reference values of attitude $\theta^*, \phi^*$ and thrust $T^*$ can be determined with Equation 2:

$$T^* = \sqrt{\mu_x^2 + \mu_y^2 + \mu_z^2} * m \tag{6}$$

$$\theta^* = \arctan(\frac{\mu_x}{\mu_z}) \tag{7}$$

$$\phi^* = \arcsin(\frac{m\mu_y}{T}) \tag{8}$$

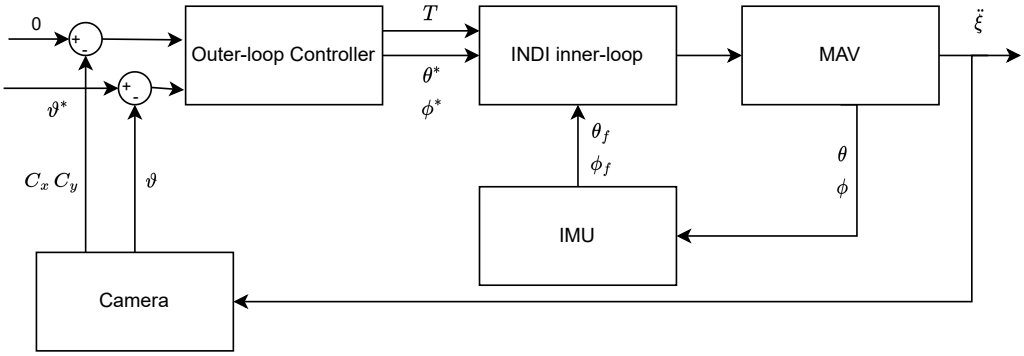


Figure 3: Block diagram of the control loops, showing the connection between the subsystems.

The set-points for attitude and thrust are fed into an inner control loop, which calculates the motor velocities of the four propellers through INDI. Here, a previously developed implementation is used from [21].

An overview of the complete control system with inner- and outer-loop can be found in Figure 3. This controller can not be implemented directly as the non-linear system dynamics require online adaptation of all controller gains in Equation 3. These gains must be proportional to the Euclidean distance to the hive $d$. The proposed solution to this problem is described in Subsection 2.4.

At some point during the landing, the quadrotor is so close to the hive the visual cues are no longer visible, and the control feedback is lost. To land, a pre-programmed landing procedure is initiated at a specific retinal size of the hive. This maneuver takes one second and starts with a forward acceleration. This is followed by a descent, after which the motors are shut down.

### 2.2 *Estimating Divergence with Adaptive Low-pass Filter*

To estimate divergence, we can use the change in size of the hive in two consecutive camera frames. To simplify the computations, we neglect perspective effects. Therefore, the hive has the same square shape from any viewing direction. Then, we get the situation depicted in Figure 4. Let the number of pixels occupied by the hive be $Q$. With a square shape, this can be converted to the length in pixels with $P = \frac{\sqrt{Q}}{2}$. When using the pinhole camera model and defining $v = \dot{d}$, the divergence can be computed with:

$$\lim_{t\to 0}(\vartheta(t)) = \lim_{t\to 0}(\frac{\dot{d}}{d}) = \lim_{t\to 0}(\frac{P_1 - P_0}{\Delta t})\frac{1}{P_0} = \frac{dP}{dt}\frac{1}{P_0} \quad (9)$$

The measurement of divergence $\vartheta$ will contain noise due to the discretization of the image in pixels. The noise depends on the change in $Q$ between two frames, which is affected by resolution, image acquisition rate, lens shape, distance $d$, and velocity $v$. The number of pixels decreases rapidly with increasing distance, making the divergence measurement deteriorate quickly.

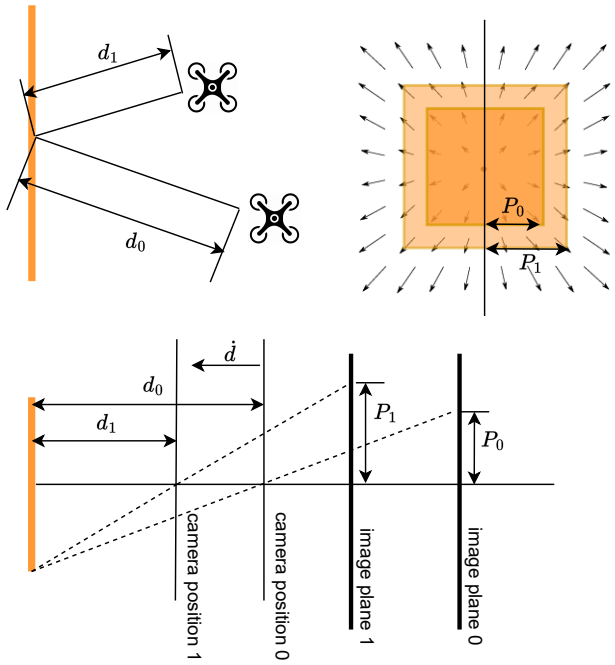


Figure 4: A schematic view of the apparent divergence of the artificial hive when flying from a position at distance $d_0$ to $d_1$. The bottom diagram shows the pinhole camera model for both positions.

Due to noise, the measurement contains high-frequency components, which can be filtered with a low-pass filter. However, there is little noise when the distance is small, and the high-frequency components become part of the actual signal due to the non-linear nature of optical flow. To deal with this, we design a first-order low-pass filter with a cut-off frequency $\omega_c$ that varies with $Q$. The continuous time transfer function of the filter in the Laplace domain is given by:

$$H(s) = \frac{1}{1 + \frac{s}{\omega_c}} \tag{10}$$

Here $\omega_c$ is calculated with:

$$\omega_c = \frac{c_{lp}}{\sqrt{Q}} \tag{11}$$

With $c_{lp}$ a scaling factor and $Q$ the number of orange pixels in the image.

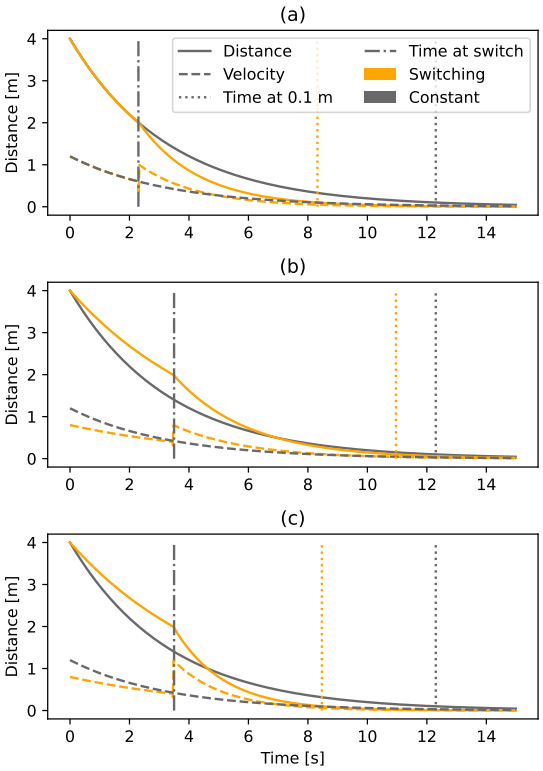


Figure 5: Comparison of the constant versus switching $\vartheta^*$ strategies. The time at which a distance of $0.1m$ from the hive is reached is marked by the vertical dotted lines. $\vartheta$ is switched halfway at a distance of 2m. In (a), $\vartheta^*$ is set to $0.3$ in the constant strategy and is switched from $0.3$ to $0.5$ in the variable strategy. In (b) $\vartheta$ is switched from $0.2$ to $0.4$. In (c), $\vartheta$ is switched from $0.2$ to $0.6$, yielding the same maximum velocity but a faster landing.

### *2.3 Divergence Set-point Switching Strategy*

In prior research, the landing strategy often consists of maintaining $\vartheta$ = $\vartheta^*$ = constant [22, 23, 13], allowing for a smooth approach, in which distance and velocity are exponentially reduced to zero. This makes the end of the landing very slow relative to the start. Inspired by new insights into the landings of bumblebees and honeybees [5, 20], an adaptive control strategy is chosen. In this strategy, $\vartheta^*$ is step-wise increased. To compare the two strategies we can look how $d$ changes with time when $\vartheta$ = $\vartheta^*$ = constant:

$$d = d_0 e^{-\vartheta^* t} \tag{12}$$

This equation is plotted for different strategies in Figure 5. In each plot, the constant $\vartheta^*$ landing is compared to a variant of the switching strategy. Figure 5(a) shows the comparison with equal initial $\vartheta^*$ and a step at half the distance. This results in a faster landing with the switching strategy.

Since a higher $\vartheta^*$ will always result in a shorter landing time, we also compare the two strategies with equal average divergence. By definition, the landing times are equal with the same time-average divergence. However, the switching strategy has a slight advantage when the divergence is averaged over the flown distance. This is shown in Figure 5(b).

Another benefit of the switching strategy is that the maximum required velocity is lowered, making it safer in cluttered environments. In Figure 5(c), the situation is shown where the maximum velocity of both strategies is equal. This again results in a faster landing with the switching strategy.

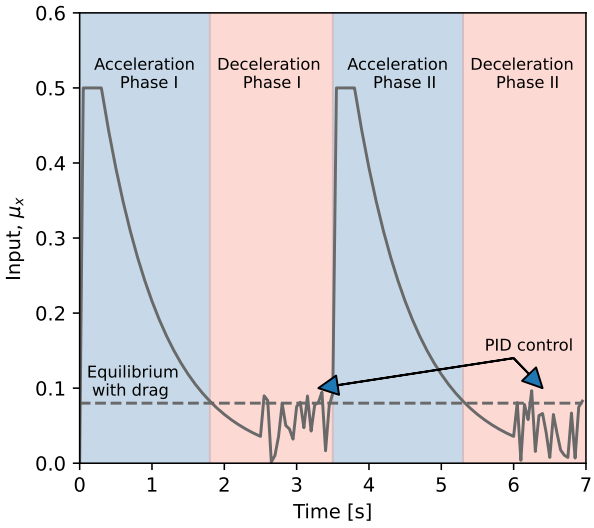


Figure 6: Divergence switching strategy. The input $\mu_x$ is plotted against time. During the first $2.5$ seconds, an input is given without visual feedback. At some point, $\mu_x$ is in equilibrium with drag, and the quadrotor starts decelerating. After $2.5$ seconds, the input is determined with PID control. This repeats until the divergence reaches a target value.

When performing the switching strategy on the quadrotor, the steps in $\vartheta$ are not instantaneous. If a new set-point is given to the controller, the delay introduced by the filter at large distances makes it challenging to achieve a good step-input response. A high gain is needed for a fast response, but the resulting overshoot can only be damped slowly due to the signal noise and delay. Therefore, we choose to give

an open-loop control input to $\mu_x$ and only monitor the divergence signal after it settles into a new set-point. The open-loop input yields reference values for $\theta$ and $T$ from Equation 6. In Figure 6 $\mu_x$ is plotted against time. The input is designed to increase $\vartheta$ quickly for a short time and then remain constant. $\vartheta^*$ is then set to the current $\vartheta$, and the controller is switched back to PID control corresponding with Equation 3. Now, the landing is characterized by two repeating phases: an acceleration and a deceleration phase. This continues until a target value of $\vartheta^*$ is reached. When this happens, the quadrotor keeps decelerating with PID control.

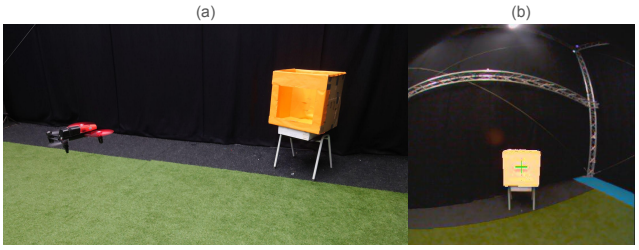


Figure 7: In (a), the experimental test setup is shown. The quadrotor has to land in the orange-painted box through the front window. The top is open, allowing light to hit the inside of the box. (b) Shows an image of the camera. The pixels detected as orange are highlighted, and a green cross-hair marks the center of the box.

### 2.4 *Scheduling the Divergence Controller Gain*

Due to the acceleration profile in the acceleration phase, there is a moment at the end of the acceleration where the thrust of the quadrotor is in equilibrium with the drag force acting on it. This point is indicated in Figure 6. At this equilibrium Equation 2 becomes:

$$\begin{bmatrix}0\\0\\g\end{bmatrix} + \frac{1}{m}\begin{bmatrix}\cos\phi\sin\theta\\-\sin\phi\\\cos\phi\cos\theta\end{bmatrix} T = \frac{1}{m}\mathbf{F}(\dot{\xi}, w) \tag{13}$$

During the landing, the thrust component in the $X_I$ direction has the greatest influence on $\vartheta$, especially during the last part of the approach when the quadrotor is centered on the hive. At this point $\theta$ is small, $\phi \approx 0$ and $T \approx g$. Then Equation 13 can be simplified to:

$$\theta g = F_x(\dot{\xi_x}, w_x) \tag{14}$$

The equilibrium velocity is only a function of $\theta$, the drag model, and the wind in the $X_I$ direction $w_x$. The initial velocity before the acceleration can be neglected. As the input to $\theta$ and the drag model are equal in every acceleration phase, the velocity at the end of the acceleration $\dot{\xi_{x_e}}$ only varies with $w_x$. Then, from Equation 1 we get:

$$\vartheta_e = c_e \frac{\dot{\xi_{x_e}}(w_x)}{\xi_{x_e}} \tag{15}$$

Where $c_e$ is a constant that depends on the drag model of the quadrotor that can be determined from flight test data. With this method, $\vartheta_e$ is only a function of wind speed and distance to the hive and can, therefore, be used to schedule the control gains in Equation 3 as follows:

$$k = \frac{c_e}{\vartheta_e} \tag{16}$$

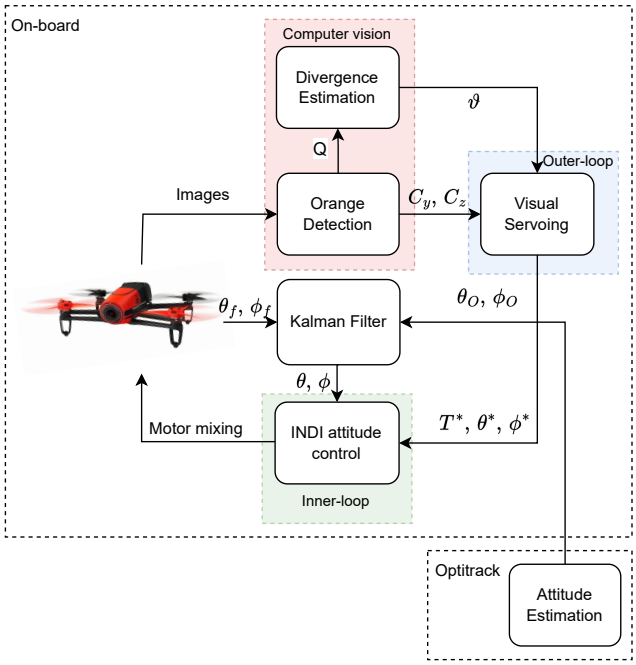


Figure 8: The modules of the autopilot software. The computer vision algorithm processes the images from the front camera of the Parrot Bebop. This calculates the center of the hive in image coordinates $C_y$ and $C_z$ and the divergence $\vartheta$. From these observations, the visual servoing module determines reference values for thrust, pitch angle, and roll angle ($T^*$, $\theta^*$, $\phi^*$). The INDI attitude control module computes the required motor velocities with feedback from the on-board IMU and the Optitrack system.

## 3 Flight Test

In this section, the results of the flight tests are presented. Two tests are performed to validate each component of the visual servoing algorithm. First, the divergence estimation is tested by tracking a sine function with the pitch angle of the quadrotor while keeping it centered on the hive. Then, the switching strategy and the visual servoing performance are validated by flying to the hive from varying initial locations and triggering the landing.

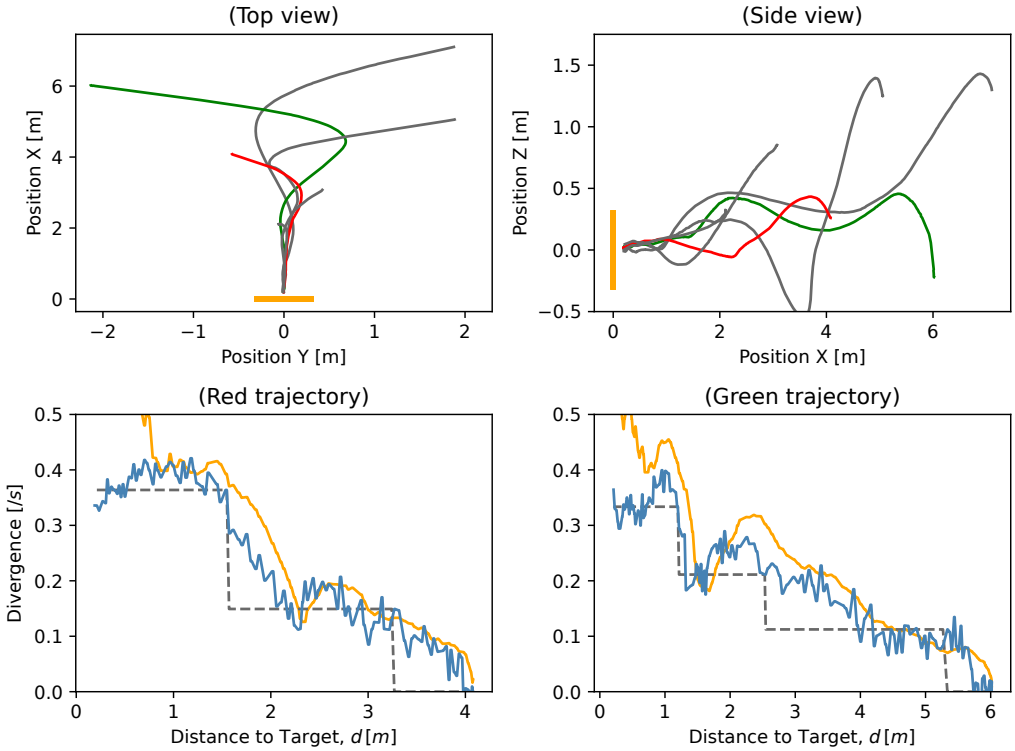


Figure 9: MAV landings from varying initial positions, top-view (a) and side-view (b). The orange lines display the orange box

### *3.1 Experimental Test Set-up*

Figure 7 shows a picture of the test setup. In the experiments, the Parrot Bebop quadrotor is used with the Paparazzi autopilot software. It is equipped with standard sensors, including a wide-angle front-looking camera and an IMU. The tests are performed in an indoor testing area with an Optitrack system that continuously tracks the position and orientation of the MAV. The position measurements are solely used to evaluate the flight performance and the attitude measurements for improving the IMU readings with a Kalman filter. This makes the tests more consistent but is not required by the visual servoing controller. In the testing area, the artificial hive is placed, which is a cube with a front window. The inside and outside of the box are painted orange to make it easily detectable by the quadrotors' camera. The top side of the box is open to light the inside of the box.

Figure 8 shows an overview of the software modules. The computer vision module processes the images from the camera. This counts the number of orange pixels $Q$ and computes the centroid in terms of image coordinates $C_y$ and $C_z$. Two subsequent images are compared to estimate divergence, and the result is filtered. Then, the visual servoing module uses these observations to run the outer control loop. The INDI module runs the inner loop with attitude feedback from the IMU and Optitrack. With a resolution of 400x400 pixels, the images are captured and processed at an average rate of 11.4 Hz during the flight tests. This average was measured over three sets of runs with five different starting locations.

### *3.2 Validation of Divergence Estimation*

To validate the divergence estimation, an experiment is performed where we let the quadrotor track a sine function with its pitch angle, and we use thrust and roll inputs to keep the quadrotor centered on the hive with the control law from Equation 3. In this way, the quadrotor repeatedly accelerates and decelerates while approaching the hive.

Figure 10 shows the measured divergence computed with Equation 9. It is important to note that the noise in the measurements is lowest close to the hive and increases with distance due to the variation in the number of pixels occupied by the hive in the image $Q$. Therefore, an adaptive filter that varies with $Q$ has been implemented. Figure 10 compares this filter to a standard low-pass filter. It can be observed that both filters have trouble tracking the ground-truth signal at large distances, but the variable filter propagates less noise. In proximity to the hive, the variable-filtered signal matches the amplitude of the actual signal better than the constant filter.

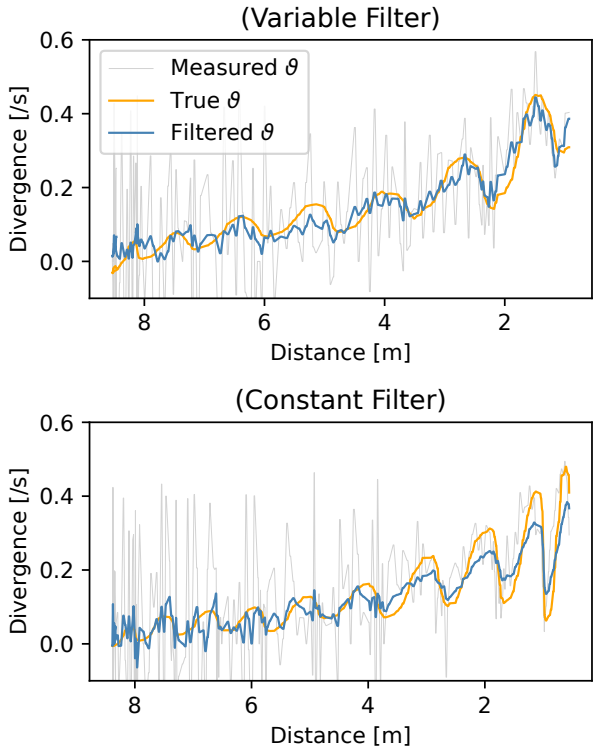


Figure 10: Comparison of the filtered $\vartheta$ measurement with variable and constant cut-off frequency. The grey line shows the unfiltered measurement, which contains substantial noise far from the hive and reduces during the approach. In both measurements the vision delay is compensated to align the measurements with $\vartheta$ obtained from Optitrack measurements. In the experiment, the quadrotor starts from a distance of $8\,m$ from the hive and is given a sine input to its pitch angle.

### *3.3 Landing with Visual Servoing*

To test the visual servoing performance, the MAV is initialized at five starting positions with varying coordinates of $\boldsymbol{\xi}$. It is decided not to include locations with large offsets in $Y$ at close distances because, in this situation, it would be reasonable to first center on the hive before approaching. This would result in the same trajectories included in the experiment.

One run is performed for each position, and the flown trajectories can be found in Figure 9. In the top view, the lateral centering response of the controller can be analyzed. Here we observe some overshoot for the trajectories with larger initial errors. This overshoot is damped during the approach. The same is true for the vertical centering response. Here, it is also visible that the quadrotor often first ascends before it centers on the hive. This is caused by an offset in the speed of the quadrotor dynamics, which is not considered in the controller's design. The vertical dynamics react faster to the controller inputs than the horizontal dynamics because it takes some time to rotate the thrust vector forward. Initially, this phenomenon is stronger than the controller's centering input.

The quadrotor is accurately positioned in front of the hive in all runs. After this, the landing is triggered at a threshold target size. The landing consists of a series of predefined maneuvers. First, the quadrotor flies forward for one second by setting $\theta = -0.03\,rad$ and $\phi = 0\,rad$. Then, it descends for half a second by reducing the thrust. Finally, the engines are switched off. In all runs, the MAV successfully landed in the orange cube while hitting the inside walls in three landings.

In Figure 9, two runs are highlighted, and their divergence is plotted against distance to show the set-point switching behavior. The steps are achieved with the open-loop switching method, shown in Figure 6. After each switch, PID control is used to keep $\vartheta = \vartheta^*$, and the proportional gain is computed from Equation 15, where $c_e = 0.9$. This way, the gain is reduced proportionally to the distance and allows for stable flight.

When comparing this to a run where a high gain is maintained in Figure 11, it can be observed that self-induced oscillations occur. De Croon showed that the onset of these oscillations is proportional to the distance and the gain of the divergence controller [17]. This can be measured by the rapid increase in variance, which could serve as an alternative threshold to trigger landing independent of target dimensions.

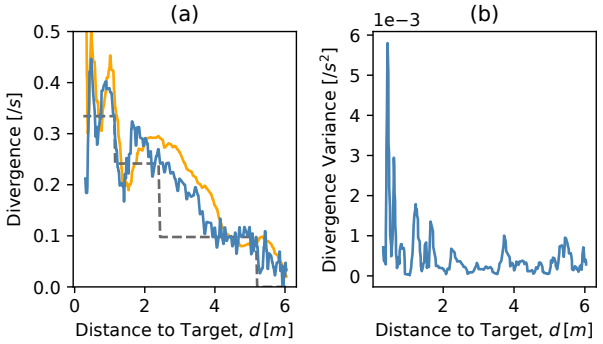


Figure 11: MAV landing with increasing $\vartheta^*$ from an initial distance of $6m$. A high gain is maintained in the divergence controller to show the self-induced oscillations when close to the hive. (a) Shows the filtered measurement of $\vartheta$ (blue line) and $\vartheta^*$ (dashed line). The true divergence (orange line) is obtained from the OptiTrack measurement system. (b) Shows how the variance of $\vartheta$ varies with distance.

## 4 Conclusion

In this study, we applied the landing strategy observed in bumblebees and honeybees to a robotic platform. It is shown analytically that this leads to a faster landing due to the increased divergence in proximity to the hive. With equal

average divergence, the switching strategy requires a lower maximum velocity than the constant divergence approach. Introducing an open-loop switching method allows for a rapid step-input response of the divergence signal, even in the presence of severe measurement noise.

To address the signal noise further, we introduced an adaptive first-order low-pass filter, dynamically adjusting based on the number of pixels occupied by the hive. This filter effectively reduces noise while passing high-frequency components of the signal near the hive. It has been demonstrated with flight tests that the visual servoing algorithm can reliably position the quadrotor in front of the hive from varying initial locations.

Moving forward, future research will focus on improving landing speed. Simple improvements can be made by increasing the spatial and temporal resolution of the camera. Additionally, the impact of wind on the visual servoing performance has to be analyzed. This will likely influence the gain scheduling in the control law. Consequently, a review of the system's stability is needed. One of the limitations of PID control is its slow disturbance rejection properties. When operating in fast wind gusts, it is likely that a different controller has to be chosen. Lastly, the triggering of the landing can be made independent of the landing targets' dimensions by implementing the stability-based distance estimation method from [17].

## REFERENCES


[1] V Kumar and N Michael. Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, 31(11):1279–1291, 8 2012.

[2] S.C. De Silva, M Phlernjai, S Rianmora, and P Ratsamee. Inverted Docking Station: A Conceptual Design for a Battery-Swapping Platform for Quadrotor UAVs. *Drones*, 6(3):56, 2022.

[3] B.E. Schäfer, D Picchi, T Engelhardt, and D Abel. Multicopter unmanned aerial vehicle for automated inspection of wind turbines. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 244–249, 2016.

[4] D Hulens, W van Ranst, Y Cao, and T Goedemé. Autonomous Visual Navigation for a Flower Pollination Drone. *Machines*, 10(5), 2022.

[5] P Goyal, E Baird, M.V. Srinivasan, and F.T. Muijres. Visual guidance of honeybees approaching a vertical landing surface. *Journal of Experimental Biology*, 226(17):jeb245956, 9 2023.

[6] M Elbanhawi, A Mohamed, R Clothier, J.L. Palmer, M Simic, and S Watkins. Enabling technologies for autonomous MAV operations. *Progress in Aerospace Sciences*, 91:27–52, 2017.

[7] N Franceschini, J.M. Pichon, and C Blanes. From insect vision to robot vision. *Philosophical Transactions of The Royal Society B Biological Sciences*, 337:283–294, 1 1992.

[8] P Corke. *Robotics, vision and control: fundamental algorithms in MATLAB, second edition*. Springer Publishing Company, Incorporated, 2nd edition, 2017.

[9] J.J. Gibson. *The perception of the visual world.* Houghton Mifflin, Oxford, England, 1950.

[10] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980.

[11] P Corke and M.C. Good. Dynamic effects in high-performance visual servoing. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 1838–1843, 1992.

[12] R Mahony, P Corke, and T Hamel. Dynamic image-based visual servo control using centroid and optic flow Features. *Journal of Dynamic Systems, Measurement, and Control*, 130(1), 12 2007.

[13] B Herissé, T Hamel, R Mahony, and F Russotto. Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow. *IEEE Transactions on Robotics*, 28(1):77–89, 2012.

[14] P Serra, R Cunha, T Hamel, D Cabecinhas, and C Silvestre. Landing of a quadrotor on a moving target using dynamic image-based visual servo control. *IEEE Transactions on Robotics*, 32(6):1524–1535, 2016.

[15] F Van Breugel, K Morgansen, and M.H. Dickinson. Monocular distance estimation from optic flow during active landing maneuvers. *Bioinspiration & biomimetics*, 9(2):025002, 2014.

[16] H.W. Ho and Y Zhou. Incremental Nonlinear Dynamic Inversion based Optical Flow Control for Flying Robots: An Efficient Data-driven Approach. *arXiv e-prints*, page arXiv:2307.02702, 7 2023.

[17] G.C.H.E. de Croon. Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy. *Bioinspiration & biomimetics*, 11(1):016004, 2016.

[18] D.N. Lee. General Tau Theory: Evolution to date. *Perception*, 38:837–850, 1 2009.

[19] B Heinrich. Resource heterogeneity and patterns of movement in foraging bumblebees. *Oecologia*, 40:235–245, 1979.

[20] P Goyal, J.L. van Leeuwen, and F.T. Muijres. Bumblebees land rapidly by intermittently accelerating and decelerating toward the surface during visually guided landings. *iScience*, 25(5):104265, 5 2022.

[21] E.J.J. Smeur, Q.P. Chu, and G.C.H.E. de Croon. Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 12 2015.

[22] H.W. Ho, G.C.H.E. De Croon, E van Kampen, Q.P. Chu, and M Mulder. Adaptive Gain Control Strategy for Constant Optical Flow Divergence Landing. *IEEE Transactions on Robotics*, 34(2):508–516, 2018.

[23] E Baird, N Boeddeker, M.R. Ibbotson, and M.V. Srinivasan. A universal strategy for visually guided landing. *Proceedings of the National Academy of Sciences*, 110(46):18686–18691, 2013.

# Part II

## Literature Review

*This part has been assessed for the course AE4020 Literature Study.

# 3

# Visually Guided Flight in Biology

In this chapter, studies into the visual perception of flying animals are explored. We start by showing a mathematical formulation describing the projection of the motion of the visible world onto the retina when moving through the environment, called optical flow, in Section 3.1. This is followed by the strategies that animals use to exploit the visual information. Firstly, the divergence strategy is described in Section 3.2 and this chapter is concluded with the time-to-contact approach in Section 3.3.

## 3.1. Optical Flow

When an observer moves relative to the surrounding scene, a moving pattern of light over the retina is perceived. This moving pattern is called optical flow and plays a crucial role in the perception of the environment by animals and humans. It gives information about the 3D structure of the scene and the ego-motion of the observer. A mathematical formulation of optical flow can be found in [4]. This makes use of the pinhole camera model. In this simplified model, the camera aperture is described as a point, and no lenses are used to focus light. For this reason, this model can not be used for cameras with a large field of view.

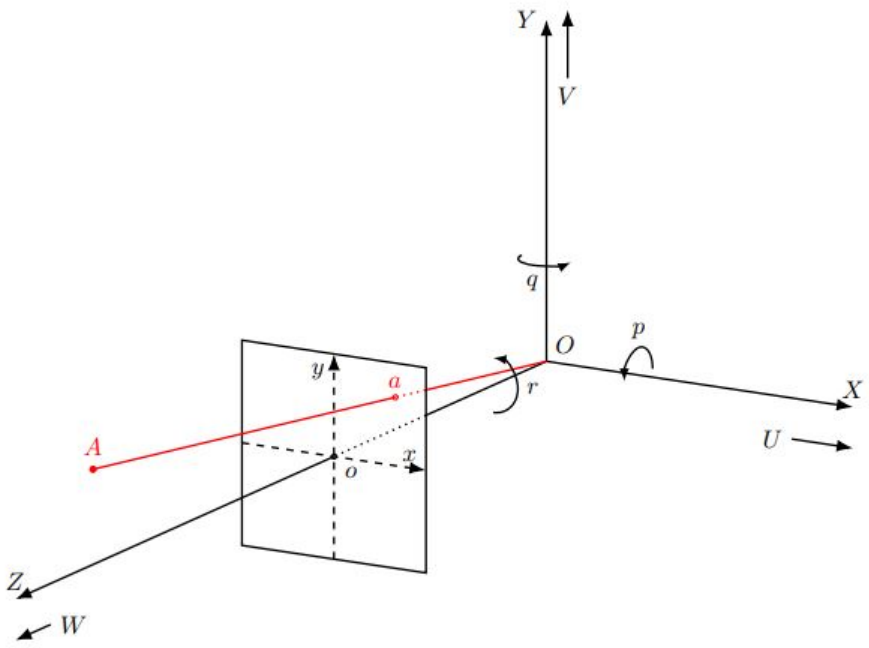


**Figure 3.1:** Pinhole Camera Model [5]

The pinhole camera model is illustrated in Figure 3.1. The observers' aperture point is located at the origin of the Cartesian coordinate system OXYZ, which remains fixed to the optical sensor with its view

direction aligned with the $Z$ axis of the coordinate system. The world point $A$ with coordinates XYZ is projected onto the image plane $(x, y)$ at point $a$. The distance between the principal point $o$ of the image plane and the observer is the focal length $f$, which is equal to $1$ for convenience.

Consider a monocular camera moving through a static scene with translational velocity $(U, V, W)$ and angular velocity $(p, q, r)$. Then the velocity components of $A$ through the moving frame are

$$\begin{aligned} \dot{X} &= -U - qZ + rY \\ \dot{Y} &= -V - rX + pZ \\ \dot{Z} &= -W - pY + qX \end{aligned} \tag{3.1}$$

The position of $p$ on the image plane can be computed from the world coordinates by dividing by $Z$.

$$(x, y) = (X/Z, Y/Z) \tag{3.2}$$

The resulting velocity of $p$ on the image plane $(u, v)$, equal to $(\dot{x}, \dot{y})$ can then be found from differentiating Equation 3.2 and substituting Equation 3.1:

$$\begin{aligned} u = \dot{x} &= \dot{X}/Z - X\dot{Z}/Z^2 = (-U/Z - q + ry) - x(-W/Z - py + qx) \\ v = \dot{y} &= \dot{Y}/Z - Y\dot{Z}/Z^2 = (-V/Z - rx + p) - y(-W/Z - py + qx) \end{aligned} \tag{3.3}$$

The velocity of the points on the image plane can be decomposed into a translational component $(u^T, v^T)$ and a rotational component $(u^R, v^R)$. By separating the optical flow components, the equations can be rewritten to:

$$u = u^T + u^R, \qquad v = v^T + v^R \tag{3.4}$$

$$with \quad u^T = (-U + xW)/Z, \qquad v^T = (-V + yW)/Z \tag{3.5}$$

$$and \quad u^R = -q + ry + pxy - qx^2, \qquad v^R = -rx + p + py^2 - qxy \tag{3.6}$$

We can define a point on the image aligned with the observers' line of motion with coordinates $x_0$ and $y_0$. The coordinates can be calculated from the observers' motion with:

$$x_0 = U/W, \qquad y_0 = V/W \tag{3.7}$$

When doing a purely translational manoeuvre or when the rotational component is subtracted from the flow field, Equation 3.5 fully describes the optical flow. Then, by substituting Equation 3.7 into Equation 3.5 we get:

$$u = (x - x_0)W/Z, \qquad v = (y - y_0)W/Z \tag{3.8}$$

From which follows that:

$$u/v = (y - y_0)/(x - x_0) \tag{3.9}$$

This shows that $x_0$, $y_0$ acts as a vanishing point of the optical flow, referred to as the focus of expansion (FOE). In the FOE, the optical flow magnitude is zero, and all other optical flow vectors point radially outward of the FOE. Additionally, if the optical flow is observed over a planar surface perpendicular to the motion direction, all points on this plane have the same distance to the observer. Let this distance be $h$. Then, by replacing $Z$ with $h$ in Equation 3.5, the optical flow of all points on this surface due to translational motion is:

$$u = (-U + xW)/h, \qquad v = (-V + yW)/h \tag{3.10}$$

Using the mathematical definition of divergence:

$$\Delta(x,y) = \frac{\delta u}{\delta x}(x,y) + \frac{\delta v}{\delta y}(x,y) \tag{3.11}$$

From this definition and Equation 3.10, the divergence $D$ of the optical flow can be calculated. When dividing by two for simplicity, the following is obtained:

$$D = \frac{W}{h} \tag{3.12}$$

This measures how fast the image is expanding outward from the FOE. The inverse of divergence is called the time-to-contact ($\tau$). It is a measure of the time it takes to close a spatial gap and is often linked to optical flow control tasks in biological studies.

$$\tau = \frac{1}{D} \tag{3.13}$$

## 3.2. Navigation with Optical Flow Divergence

Studies on the flight characteristics of insects have found that the information from optical flow is used in many flying tasks, such as obstacle avoidance, speed regulation and landing. Various experiments have been performed to study the use of optical flow for these tasks. By training insects to fly through a tunnel with varying wall patterns, the behaviour can be studied for different scenarios. In this way, honeybees have been shown to balance the optical flow on both eyes to stay in the middle of a tunnel [6]. The same behaviour is also found in bumblebees [7]. A similar experiment found that honeybees regulate their flight speed by keeping the magnitude of optical flow constant [8, 9]. This results in a higher flight speed in a large open environment and a lower flight speed in a small space. The constant optical flow strategy is not only used for speed regulation when flying through the environment but also for landing. When approaching a target, honeybees keep the optical flow divergence constant [10]. Mathematically the landing trajectory can be described with Equation 3.14. When keeping $D = k$ the distance to the target $h$ changes over time as follows:

$$h(t) = h(t_0)\, e^{-kt} \tag{3.14}$$

This shows that $h$ exponentially decreases to near zero with increasing time but never reaches it. This means that a landing action still has to be triggered at the end of the manoeuvre. It is not always the case that insects maintain a constant divergence when landing. It has been observed that bumblebees landing in a headwind maintain a high velocity until contact [11]. Furthermore, in side-wind conditions, they tend to choose a trajectory that results in a slight headwind rather than a tailwind.

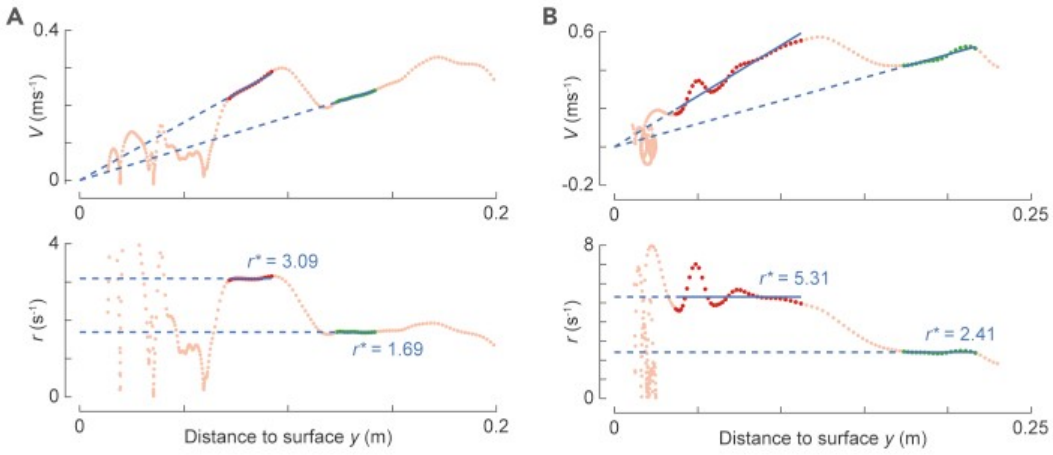


**Figure 3.2:** Landings of individual bumblebees. Velocity (V) and divergence (r) are plotted against the distance to the surface. [3]

More recently, it has also been found that, even without wind, bumblebees do not maintain a constant divergence during the entire landing. Whereas the average trajectory of multiple landings is usually provided, a study by [3] shows the behaviour in individual landings of bumblebees. Some of the observed landings can be seen in Figure 3.2. It was found that bumblebees keep the divergence only constant for a short time during their approach before switching to a different divergence set point. The set points are often increased with decreasing distance to the target. By step-wise increasing the divergence set points, a faster landing trajectory is followed that approximates a constantly increasing divergence. When keeping $\dot{D} = p$ the landing trajectory is described by:

$$h(t) = h(t_0)\, e^{-(D_0 + 0.5pt)t} \tag{3.15}$$

And the velocity by:

$$W(t) = -h(t_0)(D_0 + pt)\, e^{-(D_0 + 0.5pt)t} \tag{3.16}$$

With this strategy, $h$ still decreases asymptotically to zero, but this happens in a shorter time. This can be seen in Figure 3.3, where the trajectories of the constant divergence and the constant divergence rate control strategies are shown. Also, the required acceleration is plotted. This indicates that linearly increasing the divergence decreases landing time by a factor of two while requiring the same maximum acceleration.

How the bumblebees trigger the set point change is not yet understood. Some possibilities are the following. Firstly, the size of the target on the retina can be used as a signal to trigger the change in set points. This would be similar to using the retinal size for timing the leg extension before touchdown, found in several studies. In fruit flies, a retinal size of 60 degrees has been found to trigger the extension of the legs when landing on a flower [12]. However, this behaviour has to be learned for objects of different sizes. Secondly, a threshold value of time-to-contact ($\tau$) could be used, which is also thought to play a role in the timing of actions in animals. The drawback is that multiple velocity and distance combinations can cause a particular value of $\tau$. Therefore this information alone can not be sufficient. Lastly, when trying to replicate the constant divergence strategy on flying robots, it has been observed that oscillations occur close to the target, resulting from the singularity at $h = 0$ and the delay and noise in the observations. This phenomenon is illustrated in Figure 3.4 for different controller gains.

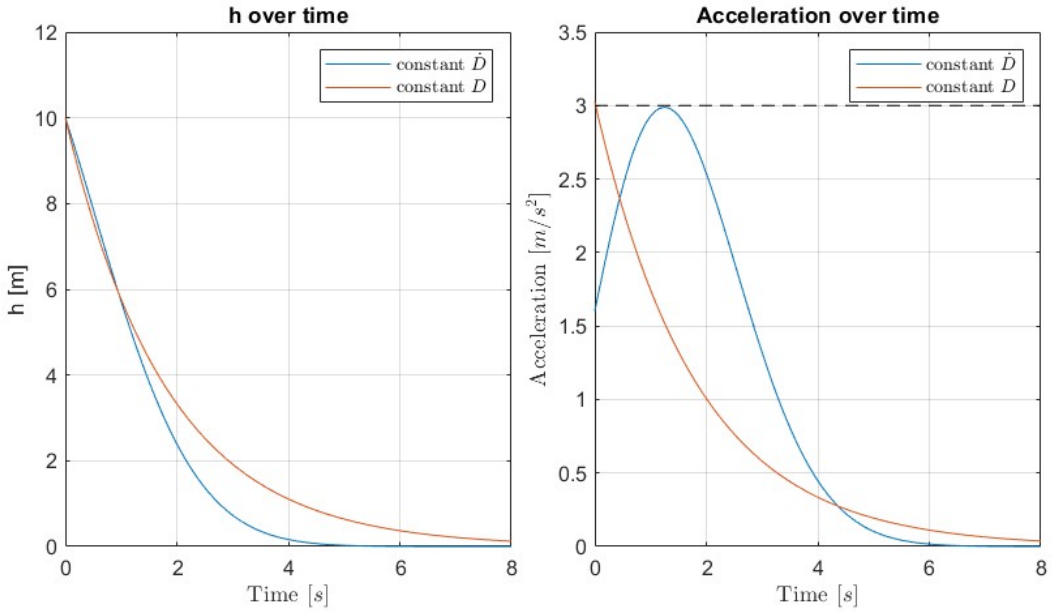


**Figure 3.3:** Comparison of the constant $D$ and constant $\dot{D}$ strategy with $p = 0.315$, $k = 0.55$ and initial conditions $W(t_0) = 4\,m/s$ and $h(t_0) = 10\,m$.

In [2], it has been found that the distance at which the oscillations occur depends on the chosen gain of the controller in the feedback loop. This is also visible in Figure 3.4. A hypothesis is that bumblebees use this strategy to match the distance to the size of an object on the retina in their learning flights, which is done in [13] for a flying robot. Another possibility is that the bumblebees continuously relate their control efforts with the observed optical flow to get a measure of the distance. This has been implemented on a flying robot in [14], which will be further explained in Chapter 4. Alternatively, the relation between the control efforts and the optical flow could be determined purely from the transient response when changing the set point. Then, the set point switch is not only used to land faster but also to make a strong input-output signal. The transient response is shown in Figure 3.2 by the yellow dots in between the constant divergence segments. Like the oscillations occurring when close to the target, the transient response might also contain hints on the distance to the target.

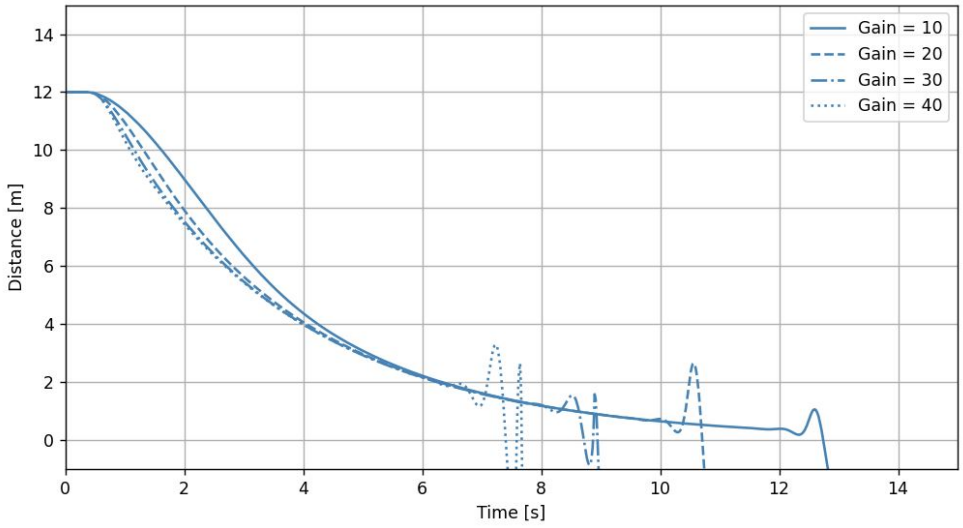


**Figure 3.4:** Divergence based landing with a time delay 0.166s and varying gain

## 3.3. Time-to-contact Control for Landing

Where insects seem to use the optical flow divergence for landing, there is substantial evidence that vertebrates use $\tau$ instead. Inspired by the work on perception of [15], Lee developed a mathematical theory using ($\tau$) [16]. This is a measure of the time it takes to close a spatial gap from an observer to a target when the approach velocity is kept constant. In practice, the approach velocity is often not kept constant but is desired to decrease with distance to the target. Lee found that many animals achieve this by keeping the derivative of $\tau$ constant when trying to close a spatial gap. This behaviour is found in drivers when braking [17], in hummingbirds for aerial docking onto a feeder [18] and in pigeons trying to land [19]. $\tau$ is also thought to play a role in the timing of actions. This can be pigeons extending their legs when close to the ground or Gannets retracting their wings when plunge-diving onto their prey [20].

When keeping $\dot{\tau} = b$ constant, the value of $b$ determines the trajectory and the velocity at the target position. This results in the following relation of the distance over time:

$$h(t) = h(t_0)\ (\frac{b}{\tau_0} + 1)^{\frac{1}{b}} \tag{3.17}$$

In Figure 3.5-B, the velocity is plotted against distance. Figure 3.5-A shows the deceleration over time. The velocity will be zero at the target when $0 < b \leq 0.5$. Values of $0.5 > b > 1$ result in a nonzero velocity, and thus a collision with the target will occur. The deceleration is constant for $b = 0.5$, displayed by the horizontal line. $0 < b < 0.5$ results in a monotonically decreasing deceleration and $0.5 > b > 1$ yields a monotonically increasing deceleration.

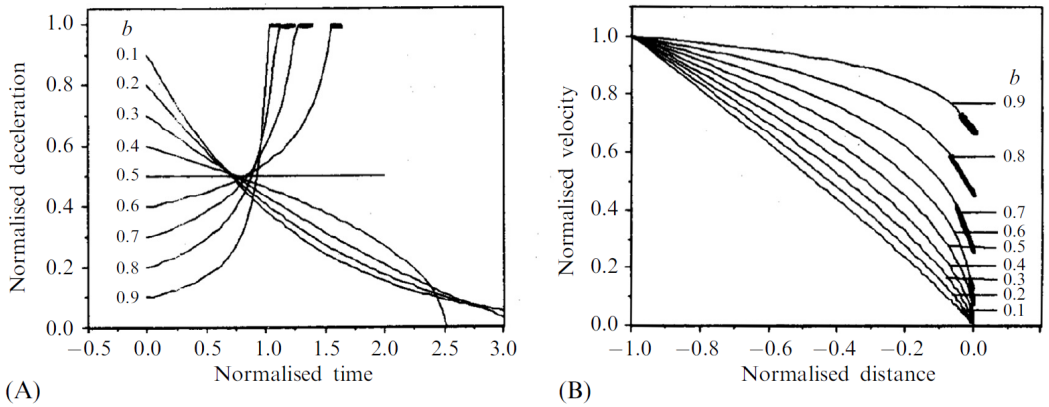


**Figure 3.5:** Kinematic profiles of an observer approaching a target while keeping $\dot{\tau} = b$ constant. [16]

An important thing to note is that, in contrast to the constant divergence strategy, keeping $\dot{\tau} = b$ constant is merely a method to control braking in front of a target. Accelerating is impossible because that implies that $\dot{\tau}$ is no longer constant. This can be explained with the example of humans braking when driving a car. When driving to a red traffic light, it must be decided when to start breaking. If the braking is initiated the braking force can be adjusted to keep $\dot{\tau}$ constant but if the driver finds out that the braking manoeuvre will take too much time, he has to abort his $\dot{\tau}$ strategy and hit the accelerator pedal. Then, he can decide again to start breaking with the $\dot{\tau}$ method.

For this reason, having the proper initial velocity at the appropriate distance is essential. When the initial velocity is too low, the manoeuvre could take a long time, or when the initial velocity is too high, the required deceleration could exceed the limit of the actuators. This also explains why the oscillations, typical to divergence landings, do not happen in the $\dot{\tau}$ approach. However, the initial decision can be difficult to make and therefore more computational capacity is needed. Therefore, it is simpler to use divergence instead when doing landings with optical flow on a robot.

# 4

# Visually guided flight in Robotics

In this chapter, a common method for controlling robots through visual cues is described, called visual servoing. This method originated from the desire to control robotic arms with visual information. This subject is largely separated from the studies of bio-inspired control strategies in the literature. However, visual servoing has also been applied to mobile robots and micro air vehicles (MAVs) in the past decade. This has required the use of other control techniques that are very similar to bio-inspired strategies.

This chapter is structured as follows. Firstly, in Section 4.1, a brief introduction to classical visual servoing techniques is given. After this, the research of visual servoing applied to MAVs is explored in Section 4.2.

## 4.1. Visual Servoing

Visual servoing can be separated into position-based visual servoing (PBVS) and image-based visual servoing (IBVS). In PBVS, the inertial pose of the observer is determined from the observed image and compared to the desired pose, from which the actuator input is calculated. However, this method is prone to camera calibration errors and needs 3D information of objects in the environment. Therefore, IBVS is often preferred. In this method, the required motion is computed directly in the 2D image coordinate system. The remainder of this section describes IBVS with the formulation from [21].

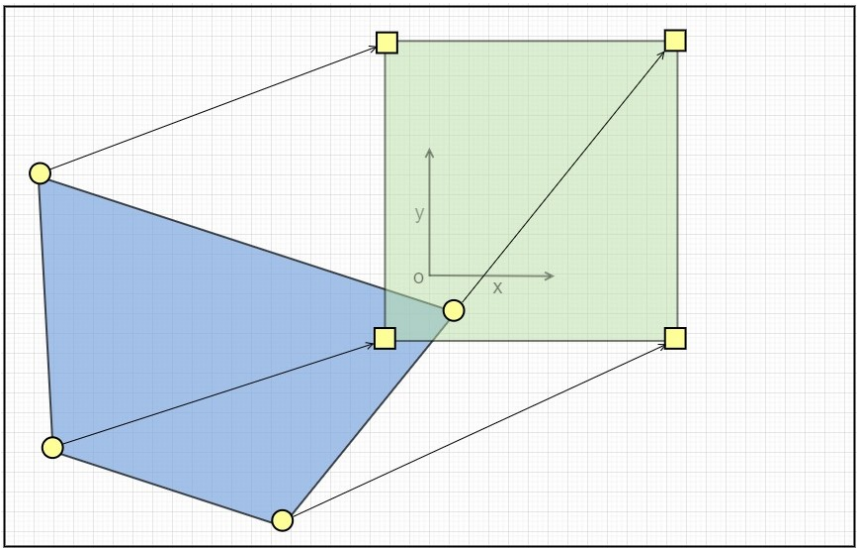


**Figure 4.1:** Image-Based Visual Servoing: Moving Image Features

IBVS aims to move features in the image to a desired location. If this is done for three or more features and there is no symmetry in the image, there will be one unique solution corresponding with the target pose in the world frame. In Figure 4.1, an abstracted IBVS problem is shown. The current feature locations are marked with circles, and the desired locations with rectangles. Designing a controller that drives the

features to the desired locations can be done by using the relation between camera motion and feature motion from the optic flow calculations in Chapter 3. We start by rewriting Equation 3.3 into matrix form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix}}_{J_p(a,Z)} \begin{bmatrix} U \\ V \\ W \\ p \\ q \\ r \end{bmatrix} \tag{4.1}$$

The matrix $J_p(a, Z)$ is often called the image Jacobian and is a function of the image coordinates of $A$ and its depth $Z$. When doing this for multiple features in the image, it follows that:

$$\begin{bmatrix} U \\ V \\ W \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} J_p(a_1, Z_1) \\ J_p(a_2, Z_2) \\ J_p(a_3, Z_3) \end{bmatrix}^{-1} \begin{bmatrix} \dot{x_1} \\ \dot{y_1} \\ \dot{x_2} \\ \dot{y_2} \\ \dot{x_3} \\ \dot{y_3} \end{bmatrix} \tag{4.2}$$

The simplest way to determine the feature velocity is to use a proportional controller. With the desired feature location $a^*$ and control gain $k$, the following control law is obtained:

$$\begin{bmatrix} U \\ V \\ W \\ p \\ q \\ r \end{bmatrix} = k \begin{bmatrix} J_p(a_1, Z_1) \\ J_p(a_2, Z_2) \\ J_p(a_3, Z_3) \end{bmatrix}^{-1} (a^* - a) \tag{4.3}$$

This method will move the camera to the desired position in space while only computations in the image domain are performed.

## 4.2. Applications to Quadrotor MAVs

Controlling an MAV using the traditional IBVS approach is challenging for three reasons. Firstly, only kinematics are controlled, and a separate control loop is needed to track the desired velocity, which requires velocity measurements. While this is easy for ground-based vehicles, aerial vehicles require heavy sensors that are computationally expensive. Secondly, controlling underactuated systems, such as quadrotors, is difficult without accounting for system dynamics, which can result in slower manoeuvres. Lastly, the control gain depends on the distance to the feature points. Therefore, to create a fast and robust controller, knowledge of the distance is required to adjust the gains during the approach.

Recent research in visual servoing has addressed these issues. In most studies, the controller design includes the system dynamics and a method to adjust the control gains. This involves controlling not only the feature velocity but also the acceleration of the features. To achieve this, an additional term is required in Equation 4.3 of the controller, which drives the feature acceleration to the desired set-point. In one study, [22], the desired feature velocity follows a pre-planned path to do perching with a quadrotor. While this approach works well with specific initial conditions, it does not generalise well to others. In another

study, [23], feature velocity is not defined beforehand. Still, there is knowledge of the absolute dimensions of objects in the environment, from which knowledge of velocity and position can be deduced.

Note that the feature velocity is equal to the optic flow at the location of the features. To reduce the effects of noise on the feature velocity measurement, it is often decided to take the optic flow of the entire target plane instead. In the remaining studies described in this chapter, the optic flow on the target plane is used as a velocity cue. To account for system dynamics, nonlinear control methods are applied, such as nonlinear dynamic inversion (NDI) and adaptive backstepping. In NDI, the relationship between the input and output of the system is used to linearise the system through differentiation. This makes it possible to design a virtual control input with conventional linear control techniques. Then the physical control input is computed from the virtual input by inversion.

NDI only works when the system model is completely known, which is rare in real-life applications. Therefore an incremental form has been developed (INDI). This uses a local linearisation of the model to compute an increment in the control input. In this increment, the internal dynamics of the system are neglected. Feedback derivatives of the system output are used to correct for internal dynamics and other disturbing factors. For a drone, this can be the rotational rates from the IMU sensors. Finally, the corrected input is used in the next control increment. The reason why INDI can be so effective is that the dynamics of the system state derivatives often react much faster to control inputs than the other states. This assumption is called time-scale separation and, unfortunately is not applicable to optical flow landings, as the flow divergence can be theoretically unbounded at small distances to the target [14].

To this end, [14] extended the INDI method to not need the full state measurements and their derivatives to compute the control increment, but only measurements of the system input and output. From past input and output measurements, terms of the system model are estimated online through least squares regression. This makes it possible to linearise the system. Then the virtual control increment is found with linear control theory and the physical input through inversion, similar to INDI. Although it has been experimentally shown that this method works for divergence-based landings with MAVs, it has a drawback. The physical control increments are computed based on the measurements of the input and output of the previous time step, but the time constants of the actuators are often larger than one time step. This leads to large actuator inputs, which makes the system less energy efficient and slower.

The study described above only looks at the one-dimensional landing problem. To extend the NDI method to the complete 3D visual servoing problem, it is usually decided to design the controller with backstepping. With backstepping, the controller is designed recursively. Starting from a small internal subsystem, the outer systems are progressively added while maintaining stability at each step. The benefit is that it is a structured way to design controllers for complex systems, and different design choices can be made for each subsystem. To ensure stability in each step, Lyapunov functions are used. With these functions, it is possible to check whether a nonlinear subsystem $\mathbf{x} = \mathbf{f}(\mathbf{x})$ is asymptotically stable. The Lyapunov function $V(x)$ is a continuously differentiable function that corresponds with the system's energy level. Geometrically the Lyapunov function describes contours around an equilibrium point. On each contour, the energy of the system is constant. If this function is defined such that it contains the entire region of the state space $\Omega$, it can be proven that the system is globally asymptotically stable with the Lyapunov stability theorem as follows:

Let $x = x^*$ be an equilibrium of the system $\mathbf{x} = \mathbf{f}(\mathbf{x})$, and $V(x)$ be a function defined on $\Omega$ with:

1. $V(x) = 0$, when $x = x^*$
2. $V(x) > 0$, for all x in $\Omega$ except $x = x^*$

It can be proven that:

1. If $\nabla V(x) \cdot f(x) \geq 0$ for all $x$ in $\Omega$ except $x = x^*$ then $x^*$ is Lyapunov stable;
2. If $\nabla V(x) \cdot f(x) < 0$ for all $x$ in $\Omega$ except $x = x^*$ then $x^*$ is asymptotically stable;
   If also $\Omega = R^n$ and $V(x) \to \infty$ as $\sqrt{x_1^2, x_2^2, ..., x_n^2} \to \infty$ then $x^*$ is globally asymptotically stable;
3. If $\nabla V(x) \cdot f(x) < 0$ for all $x$ in $\Omega$ except $x = x^*$, then $x^*$ is asymptotically stable
4. If $\nabla V(x) \cdot f(x) > 0$ for all $x$ in $\Omega$ except $x = x^*$ then $x^*$ is unstable;

One of the first studies that tried to solve the 3D visual servoing problem with the optical flow on an MAV is [24]. Here the centroid of features is used as a directional cue and optical flow for velocity. Backstepping is used to stabilise the non-linear dynamics of the MAV. However, this method still requires a measurement of the velocity of the quadrotor. Later studies have tried to relieve the requirement to measure velocity or distance. In [25], the distance is estimated during the manoeuvre. This uses the insight that the distance is a function of optic flow and the distance between the features on the image plane. This makes it possible to define a virtual function for the dynamics of the distance. It is shown that the estimated distance $\hat{d}$ from the virtual function is related to the actual distance $d$ by:

$$\frac{d}{\hat{d}} = \rho \tag{4.4}$$

Here $\rho$ is a constant that can be seen as a scaling factor, which is the information missing from the scaleless optical flow. Through backstepping, the constant $\rho$ is estimated. This estimate converges to a specific value during the approach but not the actual value. This can be seen in the simulation results in Figure 4.2. After 25 seconds $\rho$ converges to 2.3, which is different than the true value of 2.5. This happens because the error in the estimation is not optimised. The estimation is only performed to meet a total system stability criterion, the Lyapunov function.

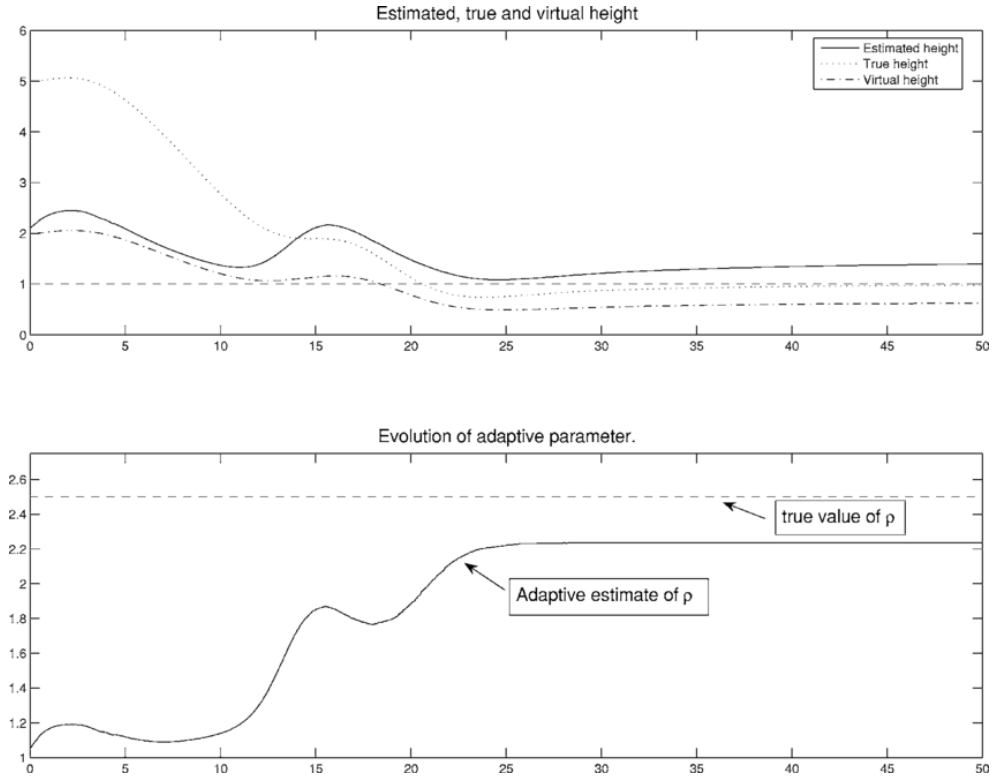


**Figure 4.2:** Simulation results of [25] with an initial distance of 5 m and an initial distance estimate of 2 m

[26] further developed this method and performed experimental tests where an MAV must land on a moving platform. The experimental results are shown in Figure 4.3. The evolution of the position of the MAV through time for all three axes is displayed. The MAV is first stabilised laterally. When doing this, the platform also only makes lateral movements. After this, the MAV is vertically stabilised over the platform, which now only moves vertically. Finally, the landing is performed on the vertically moving platform. The third plot shows the vertical position relative to the moving platform. It can be seen that during the landing, oscillations occur. This is not expected based on the mathematical analysis performed in the study, which consisted of proving asymptotic stability with the Lyapunov analysis. This is caused by the time delay in the measurements and actuation, which is not considered in the Lyapunov stability analysis. The time delay often does not cause severe problems in control systems, but it becomes problematic due to the singularity at the target position in optical flow landings. One more study using this method is [27], but here the centroid feature is slightly altered, which makes it possible for the drone to land in a specific spot on the platform. Here also oscillations occur during the landing.

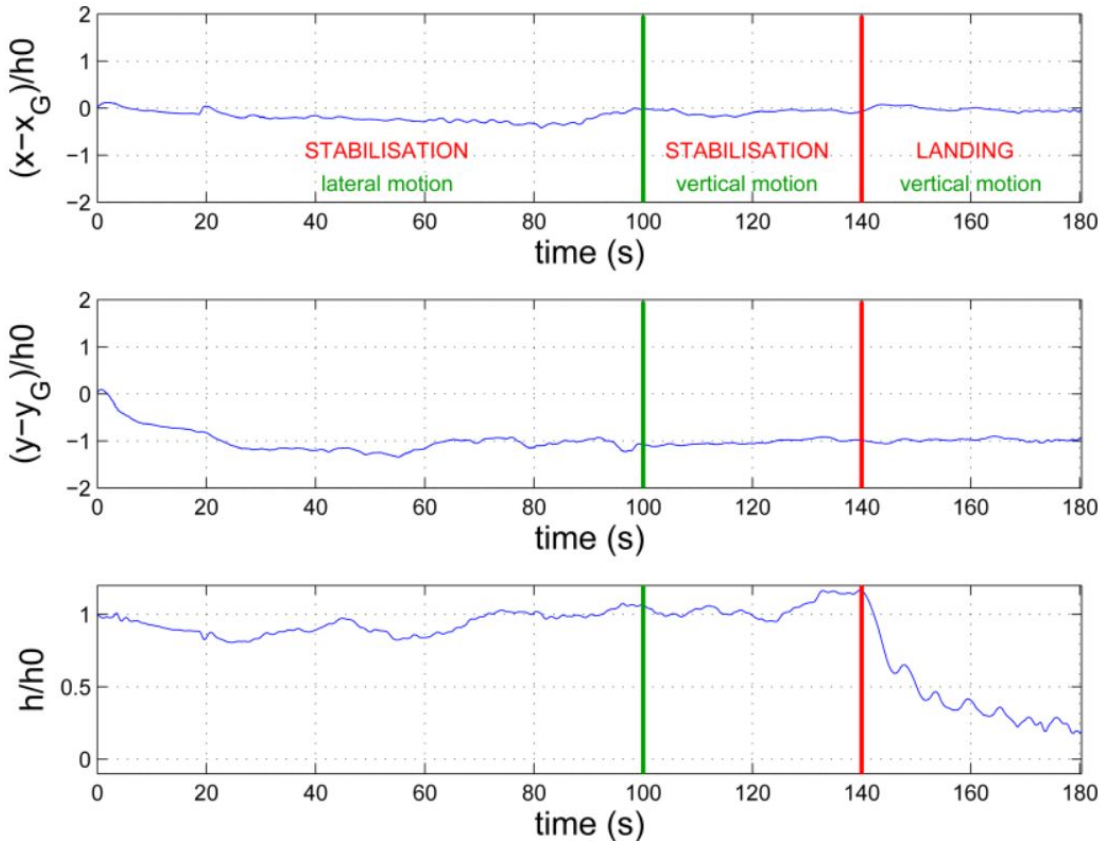


**Figure 4.3:** Experimental results of [26]

From this chapter, it can be concluded that there is still work to be done to achieve good performance with 3D visual servoing. The main problem is the lack of scale in the optic flow information. The visual servoing literature tries to solve this problem by enforcing stability through the Lyapunov stability analysis. This, however, does not consider the delays in the system; therefore, this method has suboptimal performance in experimental tests. In the experiments of [14], better performance is obtained, but this only looks at the one-dimensional landing problem, and the actuator inputs can become large due to the timing of the control increments

# 5

# Bio-inspired Control in MAVs

Inspired by the ingenious ways animals traverse their natural surroundings, roboticists have tried to mimic their behaviour. By doing so, not only the autonomous capabilities of robots are expanded, but also our understanding of the perception and behaviour of animals. Currently, most of the inspiration from nature for designing robots is drawn from insects. There are three main reasons for this. First, insects can easily be trained to fly a specific route, and large quantities of data can be gathered by tracking a lot of insects simultaneously. Another reason is their limited cognitive capacity. The observed behaviour is, therefore only a result of their low-level control systems. Finally, as the navigation skills of insects are still far ahead of their robotic counterparts, a lot can still be learned.

One of the first studies that applied insect intelligence on multiple levels in a robot was performed by [28]. Here the visual perception and behaviour of the house fly are mimicked on a ground robot. This robot has two compound eyes and a heavy computer on board. Over the past decade, it has become possible to make intelligent flying robots due to computational improvements. In this chapter, the studies into these flying robots are shown, starting with the introduction of optical flow for the perception of the environment in Section 5.1. Then, in Section 5.2, the addition of more tightly coupled perception and action is shown.

## 5.1. Optical flow control

Most of the flying tasks that have been studied in animals have also been performed with robots. In these studies, optical flow is a crucial source of information. In [29], obstacle avoidance in a corridor has been tested on a ground robot. Here the left and right peripheral flows are compared, and the robot is steered towards the side with the lowest optical flow magnitude. When a large central flow is detected, the robot turns around to avoid collision. [30] extended the corridor experiment to three dimensions by applying the optical flow balancing strategy to an MAV. Due to the more limiting mass and energy constraints of MAVs, the computational capabilities are reduced, which drives the need for more intelligent algorithms.

A different control task that is studied in both animals and robots is landing. As described in Chapter 3, two distinct strategies are recognised in animals. The constant divergence strategy, found in flying insects, and the constantly reducing time-to-contact strategy, found in birds. In [31], the constant divergence strategy is applied to a remote-controlled fixed-wing aircraft. The control system works until a certain height is reached, and a human pilot has to take over manually to prevent collision. The task of landing adds difficulty to the control problem, as the speed to the target has to be regulated. Optic flow gives only information about the velocity relative to the distance. This makes it a non-linear control problem and needs a way to adjust the gain during the approach. [32] uses a method to continuously adjust the gain of a time-to-contact controller based on the divergence. As described in the previous chapter, using time-to-contact makes the system very sensitive to the initial conditions. Therefore, additional computations are needed for this method before the landing can be initiated.

## 5.2. Sensory-motor Coordination

In robotics, most systems use passive visual observations, where sequences of images are passively gathered. This is in contrast with what we see in nature. In almost all situations, animals actively search for information by moving through their environment and observing the resulting image motion or tracking features in their visual field. This is called sensory-motor coordination or active vision and can make the

task of perception easier and can reduce computational demands. Some insects, for instance, segregate the rotational optic flow component from the translational component by using a saccadic flight and gaze strategy [33, 34, 35]. This is useful because only the translational component gives information about the structure of the environment. For example, a distant object shows less optic flow due to translation than an object that is close. When performing a pure rotational motion, the optic flow magnitude is unrelated to distance. This behaviour is mainly performed on a low level within the brain, where the motor actions are tightly coordinated with the vision sensors.

An example of a study that successfully exploits sensory-motor coordination to simplify visual perception is performed by [36]. Here an algorithm is created for an MAV that can detect gaps in the environment to fly through. The MAV hovers near the target and actively flies from side to side to make a clear depth map of the image from optic flow. From the depth map, potential gaps can be detected. Another example is found in [37]. This study addresses an inherent problem of optical flow information: the area in the visual field near the focus of expansion has a low optical flow magnitude. This can be problematic for obstacle avoidance, as the focus of expansion is aligned with the flight direction. Combining multiple translatory movements and averaging the resulting optical flow can circumvent this problem [37].

The same problem is also studied in [13]. Here the solution is found by complementing optical flow information with the size of objects on the retina. As the size of an object seen by an observer is dependent on both the dimensions of the object and the distance to the object, this yields only useful information when objects are recognised and linked to a known distance. As described in Chapter 3, the matching of the object's appearance and its distance is done through a stability-based distance estimation technique. This technique is also used in [38] to design an adaptive control strategy for landing. To adapt the control gain, a two-phase approach is used, which is shown in Figure 5.1. The gain is first increased until oscillations occur and then decreased exponentially when approaching the target. The self-induced oscillations are detected by looking at the frequency of the oscillations, as this frequency is a property of the system, similar to resonance. A commonly used method to retrieve the frequencies of a signal is the well-known fast Fourier transform. Because this is computationally expensive, the choice is made to use covariance instead. By examining the covariance function of a windowed flow divergence and a time-shifted windowed flow divergence, a single frequency can be detected. This adaptive controller allows for successful landings, even in windy conditions, but deliberately making the system unstable before initiating the landing is not desired. When looking back to the study of [14] from the previous chapter, it was found that it is also possible to monitor the input and output of the system continuously during the manoeuvre and, from this infer the distance to the target. In this way, the system does not have to become unstable to adapt the controller and make a smooth landing.

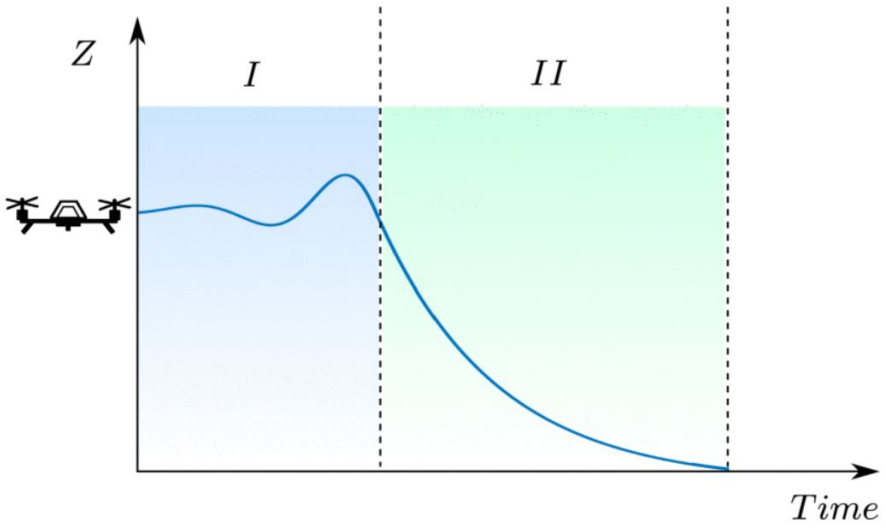


**Figure 5.1:** Two-phase landing with adaptive controller [38]

Instead of developing traditional algorithms to achieve sensory-motor coordination, this can also be

achieved through evolutionary robotics. This is a method to design intelligent systems with evolution. An overview of the evolution cycle can be found in Figure 5.2. The evolution process starts with an initial random population of agents, with each its artificial neural network coded in a genotype. Then, by iterations of evaluation, selection and variation, the agents are improved until the desired performance is reached. The benefit of this approach, unlike the more common ways to train neural networks, is that every neural network parameter can be optimised. This makes it possible to achieve better sensory-motor coordination and to design more complex neural networks, such as spiking neural networks. This type of neural network more closely mimics the biological neural networks by using asynchronous binary signals between neurons. This makes these networks more energy efficient when applied on dedicated neuromorphic hardware. Successful implementation of spiking neural networks for the optical flow landing problem can be found in [39]. The inputs to the neural network are the flow divergence and its derivative, and the output is the upward thrust. In [40], this method is further developed by running it on a neuromorphic processor, making full use of the efficiency of the spiking neural network.

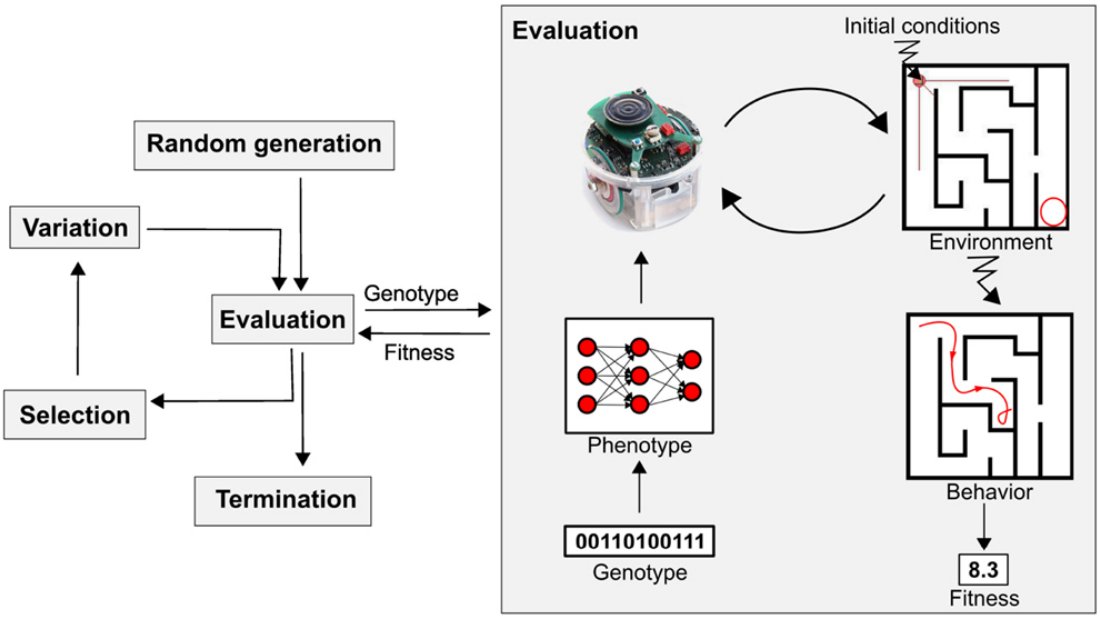


**Figure 5.2:** Evolutionary Robotics Cycle [41]

In this chapter, the research has been shown where examples of biology are applied to robotics. First, we looked at optical flow control. Here the same conclusions can be drawn as for the visual servoing literature. Namely, the scaleless property of optical flow leads to problems for both divergence and time-to-contact based controllers. Divergence landings result in oscillations close to the target, and time-to-contact landings are very sensitive to the initial conditions. Secondly, it was described how it can be beneficial to actively gather information from the environment, which animals almost always do. Several studies have been shown that use this very effectively for robots. Lastly, evolutionary robotics has been introduced, which is a suitable design approach to achieve good sensory-motor coordination and can also be used to make energy-efficient neural controllers. This makes it an interesting method to include in the next phase of the thesis.

# 6

# Literature Synthesis

This chapter provides a synthesis of the literature collected for the study. This literature study aimed to find the current state of the research and show opportunities for further work. Additionally, this study was used to gain insights into the relevant topics for the thesis. Firstly, biological studies into flying animals were gathered. This showed that optical flow plays a vital role in the perception of the environment. However, the ways animals use this information differs. Secondly, the most used methods for the control of robots with cameras were explored. Thirdly, it was shown how the biological findings could be used effectively for the perception and control of MAVs. Finally, the main problems were discussed, and some solutions available in the literature were shown.

Optical flow contains information about the structure of the environment and the motion of the observer [15]. This information plays a crucial role in biology for the perception of the environment, where it is used for many navigational tasks by flying insects and birds. Landing is a task where animals rely heavily on optical flow information. Here two main strategies have been observed. Firstly, honeybees keep the rate of expansion of the image constant when landing on a vertical platform [10]. Fruit flies use the same strategy when approaching a target and extend their legs at a fixed retinal size [12]. Secondly, pigeons use the inverse of the expansion rate, also known as time-to-contact, to decelerate when landing [19]. The pigeons keep constant the rate of change of the time-to-contact and extend their legs at a fixed time-to-contact to the surface. This behaviour is also found in many other birds and generally results in faster landings than the insect strategy. However, recent studies suggest that insects might use a strategy that matches the bird strategy more closely. When looking at individual landings of bumblebees, the expansion rate is kept constant for a short time and is increased step-wise when approaching a target [3]. This strategy approximates the bird strategy and is comparable in speed.

Attempts have been made to implement the biological findings in MAVs. This showed that optical flow information alone is insufficient to perform smooth landings. This is caused by the fact that optical flow gives information about the ratio of velocity and distance, but both can not be determined absolutely. For this reason, every application of optical flow control in robots needs additional information to obtain an estimate of velocity or distance. Without this information, the system can become unstable near the target for divergence-based controllers [2], for time-to-contact based controllers problems arise at the start of the manoeuvre, because this strategy depends highly on the initial conditions [32].

Additional information about the velocity of the robot or its distance to a target can be obtained by using more sensors. Some examples are the following. Tachometers can be used for robotic arms to estimate the movement of the joints [42]. [43] uses sonar to land a quadrotor on a moving target. An additional camera can estimate distance with epipolar geometry [44]. Also, GPS can be used to get positional and velocity information. All the extra sensors add complexity, weight and energy requirements to the system. For small flying robots, this can be problematic. However, in nature, most flying animals can perform very complex tasks with almost solely visual information. Therefore mimicking their behaviour could lead to efficient control strategies.

Some research is available that gathers the additional information without adding extra sensors. In [2], the distance to objects is estimated through the stability of the control system. The distance to an object was found to be proportional to the gain of a divergence-based controller. This strategy is also used in [13] for learning to match an object's distance with its visual appearance. This shows the potential of the

stability-based distance estimation, and it is hypothesised that this is also a mechanism insects use to land. For the first learning flights, it is still useful to have a fast and robust strategy that does not use prior knowledge of the environment, which will be the focus of this thesis. An example of such a strategy can be found in [14]. Here an extension of INDI is used, a commonly used technique for nonlinear control problems. The control inputs are based on the relation between the previous inputs and outputs of the system. Due to the discrete steps of the control increments that can be smaller than the time constants of the actuators, large actuator inputs can occur. Alternatively, a learning-based approach can be used for vertical landing with an MAV [40]. A spiking neural network is ran on neuromorphic hardware, which makes it very energy efficient. Here the divergence is fed into the neural network, and the thrust set-point is the output.

The above research only studied the control of MAVs in one direction, towards the target. This can be seen as closing a spatial gap from the observer to the target. To land in a specific location, the spatial gaps in all three dimensions must be closed simultaneously. [32] Developed a method to do this with a time-to-contact based controller. To make the system less sensitive to initial conditions and improve the performance close to the target position, continuous gain scheduling is used that is proportional to the divergence. This improves the sensitivity to initial conditions but remains insufficient for consistent fast landings.

Most research on the 3 Dimensional control of MAVs originates from the visual servoing literature. Here, the optical flow is often used as a velocity cue and the centroid of image features as a directional cue [24, 25, 27, 45, 26]. Non-linear adaptive control techniques are applied to achieve asymptotic stability during the entire manoeuvre, which uses the system model. The controller gains are tuned based on the distance to the target, estimated during the maneuver. However, the estimation does not converge to the true value because it is only estimated to meet a total system stability criterion. In these studies, extensive stability analyses are performed based on Lyapunov theory, but this does not consider the delay in the observations occurring in the real world. Therefore in experimental tests, the performance is sub-optimal.

# 7

# Conclusion

In this literature review, the research into visually guided flight has been explored. The studies of flying robots show that a bearing-only sensor, such as a monocular camera, is insufficient to perform a smooth landing in the whole flight domain. In some studies, additional sensors are used, such as sonar in [43]. Others use prior knowledge of the environment [23]. Another method is to use adaptive controllers to deal with the non-linearity of the system. This can be purely based on the system output [32] or on the identification of the model of the system [25, 26, 14], which is used in most 3-dimensional visual servoing studies. The most important parameter in the model that must be estimated is the distance to the target, as the gain of the controller depends on it. In the visual servoing literature, the parameters are often estimated based on the stability analysis of the system with Lyapunov methods. However, the time delay of the observations and actions is not considered, resulting in poor performance in experimental tests. In [14], better experimental results are obtained, but only one degree of freedom is controlled.

When looking at the biological studies, there is evidence for an adaptive control strategy in bumblebees [3]. How the adaptations are triggered is, however, not yet known. A possibility is the use of the size of objects on the retina, as it has been found in many studies that the retinal size of objects is used for the timing of actions. To obtain relevant information from the retinal size, it must be matched with the associated distance in the initial learning flights. So far, this learning process has not been studied in these experiments, but the principle has been successfully applied to flying robots [13]. However, it remains useful to make fast and accurate landings without prior knowledge of objects in the environment. For this, a different source of information has to be found for the adaptation of the controller.

The initial landings controlling all degrees of freedom will be the main focus of the thesis. An adaptive control strategy will be developed for this, mimicking the behaviour found in bumblebees. To adapt the controller during the approach, a method to estimate the distance to the target has to be found. [2, 14] show that this information is present in the coordination between actions and perception. The key challenge will be to find a practical way to estimate the distance with delay and noise in the observations while controlling the MAV in three axes. The task of adaptation of the controller is well suited for a learning approach, which is a direction that can be explored in the next phase of the thesis. The resulting algorithm will be compared to the existing literature based on speed, sensitivity to initial conditions and system model complexity.

# Part III

## Additional Results and Closure

# 8

# Time-to-contact for Landing

As described in the Chapter 3, there is an alternative landing strategy when using optical flow information. Instead of divergence, its inverse (time-to-contact) can also be used to decelerate in front of a target. This strategy is commonly observed in larger animals, such as birds. In this chapter, a further explanation is given for the discrepancy between the insect and bird strategies.

The time-to-contact is the inverse of divergence and is therefore given by:

$$\tau = \frac{d}{v} \tag{8.1}$$

In the bird strategy, the rate of change of $\tau$ is kept at a negative constant $b$. Then, a simple proportional controller can be implemented:

$$\dot{v} = k * (\dot{\tau} - b) \tag{8.2}$$

With this controller, the distance to the target changes over time as follows:

$$d(t) = d(t_0)\ (\frac{b}{\tau_0} + 1)^{\frac{1}{b}} \tag{8.3}$$

This has a great benefit over the insect strategy, as it allows for a non-zero final velocity by choosing $0.5 < b < 1$. This results in a firm landing and eliminates the singularity at $d = 0$. However, as pointed out in Chapter 3, this strategy faces a problem when $v = 0$, i.e., when hovering. Therefore, a maneuver has to take place before the landing can be initiated. The landing speed is now dictated by the initial velocity and distance, requiring precise control over the initial velocity to make a fast landing. This parameter is not directly measurable from vision. Even with an optimal initial velocity, any external disturbances can change the velocity during the landing and deteriorate the landing performance. For birds, this often does not lead to problems because the acceleration they experience due to gravity is much stronger than that from wind. However, insects experience large accelerations from wind, as they have low Reynolds numbers and low weight. If an insect were to use the bird strategy in strong wind gusts, its velocity would change rapidly. When the velocity crosses zero, $\tau$ becomes unbounded due to the singularity at $v = 0$. This leads to large control inputs and a potentially unstable control system.

To summarize, the bird strategy has benefits over the insect strategy but is only suitable if the drag-to-weight ratio is low and the maneuvers have high momentum. It also requires a better understanding of the environment, as the initial velocity has to be controlled before initializing the landing.

# 9

# Estimating Distance with Set-point Switching Landing Strategy

The scientific article shows how the switching divergence strategy of bees can improve the landing performance. However, this improvement is due to the increased divergence during the landing, a change that could also be achieved gradually rather than in discrete steps. This chapter investigates the possibility that the discrete steps serve as a method to estimate the distance to surrounding objects. We conduct simulated landings with a quadrotor to explore this hypothesis. Accurate distance information is of great value because it is needed to adapt a divergence-based controller and can be used to trigger task-specific actions.

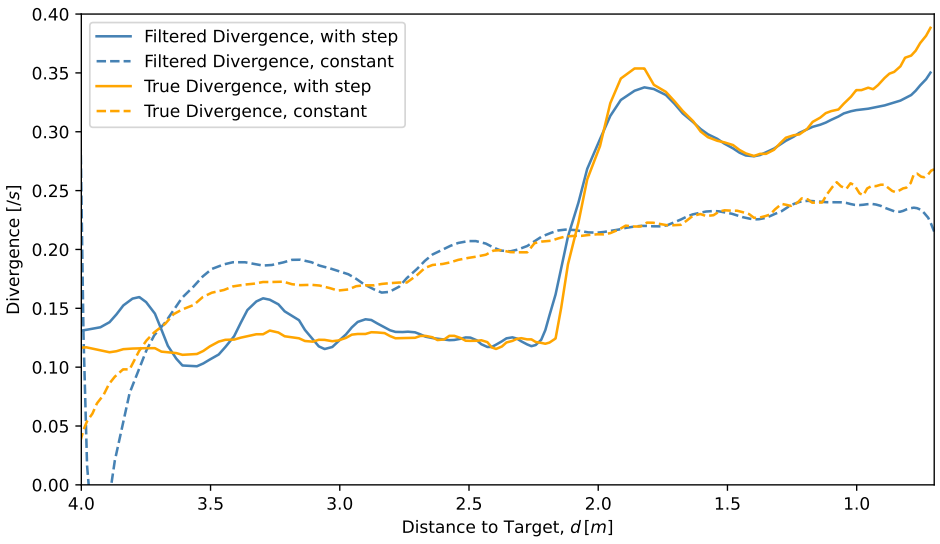


**Figure 9.1:** Two Simulated landings with a quadrotor. The constant divergence strategy is compared to the switching strategy. In the switching strategy, the quadrotor accelerates to the new set-point at 2.2 meters from the landing target.

For our analysis, we go back to the definition of the optical flow divergence $\vartheta$, which is a measure of the ratio of velocity $v$ and distance $d$ relative to other objects. Differentiation of divergence gives:

$$\frac{d}{dt}\vartheta = \frac{d}{dt}\frac{-v}{d} = -\frac{\dot{v}}{d} + \frac{v^2}{d^2} = -\frac{\dot{v}}{d} + \vartheta^2 \tag{9.1}$$

By rewriting, we obtain a relation for the distance:

$$d = \frac{\dot{v}}{\vartheta^2 - \dot{\vartheta}} \tag{9.2}$$

This relation provides a general method to estimate distance for any accelerating observer and is used in [46] and [47]. With perfect measurements, the numerator and denominator of Equation 9.2 go to zero simultaneously. However, in practice, it can be hard to accurately determine the acceleration $\dot{v}$ because on-board inertial measurement units have imperfections that cause drift, and the vibrations from the rotors create noise in the measurements. Moreover, taking the derivative of the noisy divergence signal can be very inaccurate. Therefore, it can not be guaranteed that Equation 9.2 remains bounded. To improve the distance estimate, keeping the acceleration and, therefore, the denominator far away from the singularity at $\dot{\vartheta} = \vartheta^2$ is beneficial.

This is where the discrete switching strategy can help, as higher acceleration and deceleration occur during and after the switch. In Figure 9.1, the divergence is shown for the constant and switching strategies. When the step input to the divergence is given, the quadrotor first accelerates quickly to the new divergence set-point and subsequently decelerates to maintain that divergence.

To show the difference between the two strategies, we use Equation 9.2 to calculate the distance to the landing target from the measured divergence and the true acceleration obtained from the simulation. Then, the error with the ground truth distance is determined for both strategies and can be found in Figure 9.2. In the run of the switching strategy, the error is about 1 meter during the acceleration and increases rapidly during the transition to deceleration, as $\dot{\vartheta} \approx \vartheta^2$. After this, the quadrotor decelerates, and the error remains below 0.5 meters, which shows an improvement over the constant strategy.

Note that the estimates are very poor for both strategies at distances greater than 2.5 meters. This will only worsen when reading the acceleration from an IMU instead of using the true acceleration. This is in line with the findings in the scientific article, where it was shown that the noise in the divergence measurements increases with distance. This means that methods from prior research, using Equation 9.2, are likely to fail when used to land on a tiny target, from which it is challenging to obtain accurate divergence measurements.

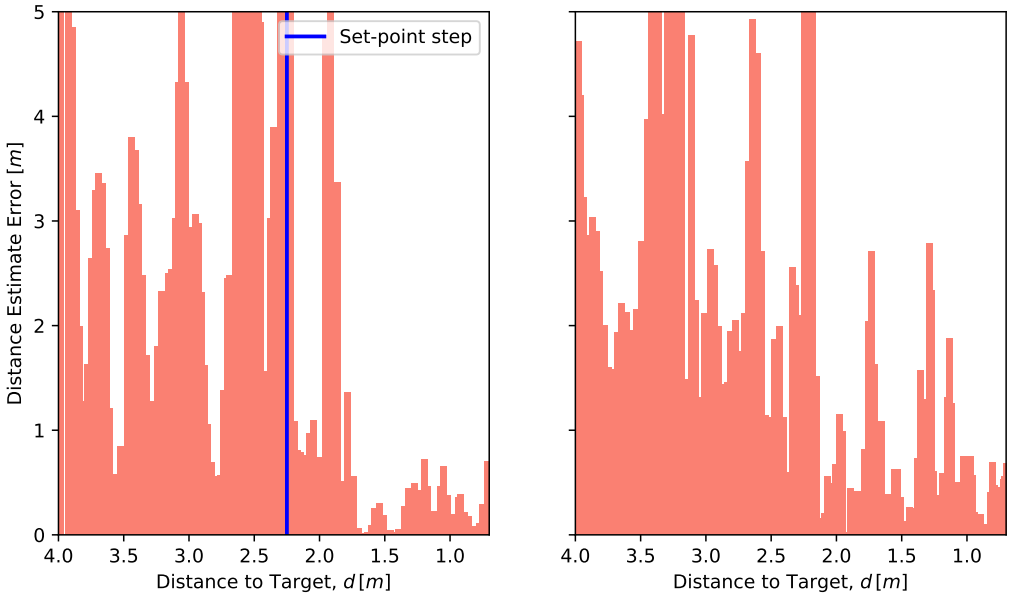


**Figure 9.2:** Distance estimation error

# 10

# Conclusion

Optical flow control has proven to be an effective way for animals and machines to traverse their environment. However, there are still challenges to overcome before it can be widely adopted to control small flying robots. Solutions can be found in nature, where we see optical flow control being used by animals to perform all sorts of tasks reliably. This research has led to new insight into the landing behavior of bumblebees, which shows an adaptive strategy where the optical flow expansion of the landing target is step-wise regulated. To see how this can improve existing optical flow control methods for the task of landing, the following research objective was formulated:

**Research Objective**

Developing a bio-inspired visual servoing strategy, based on the behavior found in bumblebees, to perform landings with a quadrotor MAV, using only a monocular camera and an inertial measurement unit

To this end, we developed an open-loop switching method, recognizing that the divergence estimates are particularly noisy at large distances. To improve the divergence estimation, an adaptive filter is introduced that considers the size of the visual servoing target. An adaptive control law is used to deal with non-linear system dynamics, where the control gain is scheduled based on the effectiveness of the actuator inputs to increase the divergence during the steps in divergence.

It is demonstrated with flight tests that the quadrotor can reliably land on the target from varying initial positions. The introduction of the adaptive filter improved the accuracy of the divergence estimation in proximity to the visual servoing target, compared to a standard low-pass filter, while reducing the noise at greater distances. However, it still failed to track the baseline divergence signal at these distances. Due to the high-frequency noise and the low sampling rate of the captured images, filtering out the noise without introducing large delays remains problematic. Analytically, it is shown how the switching strategy slightly reduces landing time compared to a constant divergence strategy with the same average divergence over distance. This strategy also reduces the maximum velocity that is required during the landing.

## Recommendations

The main objective of this thesis is achieved, but significant improvements still have to be made in the speed and robustness of the landing before it can be used in practice. Moreover, tests must be performed in more challenging environments, including varying lighting conditions and wind. Changes likely have to be made to the method to work in these environments. This section aims to provide recommendations to improve the proposed method, which can be used in future research, with the knowledge acquired throughout the thesis.

**Recommendations to improve speed and robustness of visual servoing**

- **Divergence step response**: In the developed method, the changes in divergence take 2.5 seconds. During this time, the inputs to the forward acceleration do not use feedback, making the controller vulnerable to disturbances. This time should be reduced to increase the robustness and speed

of the landing. A bumblebee can do the set-point switches much faster than the quadrotor used in the experiments because it has much lower inertia. Therefore, the switching strategy might be more suitable for a smaller MAV. An additional way to decrease the time of the step response is to use optimal control strategies that take into account the complex quadrotor dynamics during rapid accelerations. Here, machine learning can give great performance benefits, substantiated by the recent advancements in machine learning for drone racing.

- **PID controller**: A shortcoming of PID control is its slow rejection of disturbances. This is particularly problematic for landings in outdoor environments, as the MAV flies close to objects with disturbances from wind. Incremental non-linear dynamic inversion is proven to be much more effective in rejecting disturbances [48] but requires accurate measurements of both divergence and the quadrotors' accelerations. This thesis showed that this only holds in proximity to the observed target. Therefore, either the measurements must be improved, or we should only use them for feedback when the target has a sufficiently large size in the images.

**Recommendations to improve divergence estimation**

- **Increasing velocity**: When flying faster, the change in the number of pixels occupied by the target is larger between two images. This improves the signal-to-noise ratio of the divergence measurement.
- **Image acquisition rate**: The Parrot Bebop quadrotor used in the experiments was first released in 2014. Since then, the computational capabilities of small electronics have seen great improvements. Therefore, a higher image processing rate should be possible on a newer MAV. The higher sampling rate makes removing the high-frequency noise from the measurements easier. In addition, improvements can be made to the efficiency of the computer vision algorithms.
- **Image resolution**: Similar to improving the temporal resolution of the measurements, improving spatial resolution also reduces measurement noise. Since the noise is particularly problematic when the target is small in the image, an improvement can be made by focusing only on the region where the target is observed. This allows for a higher resolution of the important features without adding computational load.

**Recommendations for the optical flow strategy**

- **Time-to-contact versus divergence for landing**: The decreasing time-to-contact landing strategy has an advantage due to the ability to land with a non-zero velocity but is sensitive to changes in velocity during the descent. Therefore, the divergence-based strategy is recommended for small flying robots in windy environments.
- **Discrete versus continuous change in divergence**: During the flight test experiments of this thesis, we recognized the challenge of achieving fast step input responses to divergence. Consequently, a continuously increasing divergence might offer better performance. However, this approach would also remove the large accelerations around the set-point change, which can be used to obtain distance estimates. More experiments must be performed to validate the accuracy of the distance estimate with the set-point switching strategy.

**Recommendations to broaden the scope of the research**

- **Optical flow algorithm**: In the experiments performed during this thesis, a simplified method was used to determine the divergence by counting the number of orange pixels. A more general method is to find correspondences between pixels in two subsequent images and determine optical flow from their displacement. Efficient algorithms exist that can replace the currently used method.
- **Challenging environmental conditions**: The flight tests were performed in an indoor lab setting with predictable lighting conditions and no wind. Additional tests should be performed in more challenging conditions.
- **Machine Learning**: The task of landing is simple enough to be performed through traditional algorithms. However, this task is not done in isolation but is part of a larger process to reach a specific goal. When multiple tasks have to be performed in succession, more complex decisions must be made that can be more suitable for a machine learning approach.

# References


[1] 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”. In: *International Journal of Micro Air Vehicles* 1.2 (2009), pp. 71–97.

[2] G.C.H.E. De Croon. “Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy”. In: *Bioinspiration & biomimetics* 11.1 (2016), p. 016004.

[3] P Goyal et al. “Bumblebees land rapidly and robustly using a sophisticated modular flight control strategy”. In: *iScience* 24.5 (May 2021), p. 102407. DOI: 10.1016/J.ISCI.2021.102407.

[4] H.C. Longuet-Higgins and K Prazdny. “The interpretation of a moving retinal image”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 208.1173 (1980), pp. 385–397.

[5] J.J. Hagenaars. “Evolved Neuromorphic Control for High Speed Divergence-based Landings of MAVs”. PhD thesis. Delft: Delft University of Technology, 2020, p. 34.

[6] M Srinivasan, M Lehrer, W Kirchner, and S Zhang. “Range perception through apparent image speed in freely flying honeybees”. In: *Visual neuroscience* 6 (June 1991), pp. 519–535. DOI: 10.1017/S095252380000136X.

[7] J.P. Dyhr and C.M. Higgins. “The spatial frequency tuning of optic-flow-dependent behaviors in the bumblebee Bombus impatiens”. In: *Journal of Experimental Biology* 213.10 (May 2010), pp. 1643–1650. DOI: 10.1242/jeb.041426. URL: https://doi.org/10.1242/jeb.041426.

[8] E Baird, M.V. Srinivasan, S Zhang, and A Cowling. “Visual control of flight speed in honeybees”. In: *Journal of Experimental Biology* 208.20 (Oct. 2005), pp. 3895–3905. DOI: 10.1242/jeb.01818. URL: https://doi.org/10.1242/jeb.01818.

[9] M.V. Srinivasan, S Zhang, M Lehrer, and T Collett. “Honeybee Navigation en route to the Goal: Visual Flight Control and Odometry”. In: *The Journal of experimental biology* 199 (Feb. 1996), pp. 237–244. DOI: 10.1242/jeb.199.1.237.

[10] E Baird, N Boeddeker, M.R. Ibbotson, and M.V. Srinivasan. “A universal strategy for visually guided landing”. In: *Proceedings of the National Academy of Sciences* 110.46 (2013), pp. 18686–18691. DOI: 10.1073/pnas.1314311110. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1314311110.

[11] J.J. Chang, J.D. Crall, and S.A. Combes. “Wind alters landing dynamics in bumblebees”. In: *Journal of Experimental Biology* 219.18 (Sept. 2016), pp. 2819–2822. DOI: 10.1242/jeb.137976. URL: https://doi.org/10.1242/jeb.137976.

[12] F van Breugel and M Dickinson. “The visual control of landing and obstacle avoidance in the fruit fly Drosophila melanogaster”. In: *The Journal of experimental biology* 215 (June 2012), pp. 1783–1798. DOI: 10.1242/jeb.066498.

[13] G.C.H.E. De Croon, C De Wagter, and T Seidl. “Enhancing optical-flow-based control by learning visual appearance cues for flying robots”. In: *Nature Machine Intelligence* 3.1 (2021), pp. 33–41. DOI: 10.1038/s42256-020-00279-7. URL: https://doi.org/10.1038/s42256-020-00279-7.

[14] Y Zhou, H.W. Ho, and Q Chu. “Extended incremental nonlinear dynamic inversion for optical flow control of micro air vehicles”. In: *Aerospace Science and Technology* 116 (2021), p. 106889. DOI: https://doi.org/10.1016/j.ast.2021.106889. URL: https://www.sciencedirect.com/science/article/pii/S1270963821003990.

[15] J.J. Gibson. *The perception of the visual world.* Oxford, England: Houghton Mifflin, 1950, pp. 242, xii, 242–xii.

[16] D.N. Lee. “General Tau Theory: Evolution to date”. In: *Perception* 38 (Jan. 2009), pp. 837–850.

[17] D.N. Lee. “A Theory of Visual Control of Braking Based on Information about Time-to-Collision”. In: *Perception* 5.4 (Dec. 1976), pp. 437–459. DOI: 10.1068/p050437. URL: https://doi.org/10.1068/p050437.

[18] D.N. Lee, P.E. Reddish, and D.T. Rand. “Aerial docking by hummingbirds”. In: *Naturwissenschaften* 78.11 (1991), pp. 526–527. DOI: 10.1007/BF01131406. URL: https://doi.org/10.1007/BF01131406.

[19] D.N. Lee, M.N.O. Davies, P.R. Green, and F.R. Van Der Weel. “VISUAL CONTROL OF VELOCITY OF APPROACH BY PIGEONS WHEN LANDING”. In: *Journal of Experimental Biology* 180.1 (July 1993), pp. 85–104. DOI: 10.1242/jeb.180.1.85. URL: https://doi.org/10.1242/jeb.180.1.85.

[20] D.N. Lee and P.E. Reddish. “Plummeting gannets: a paradigm of ecological optics”. In: *Nature* 293.5830 (1981), pp. 293–294. DOI: 10.1038/293293a0. URL: https://doi.org/10.1038/293293a0.

[21] P Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB, Second Edition*. 2nd. Springer Publishing Company, Incorporated, 2017.

[22] J Thomas, G Loianno, K Sreenath, and V Kumar. “Toward image based visual servoing for aerial grasping and perching”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 2113–2118. DOI: 10.1109/ICRA.2014.6907149.

[23] J Thomas, G Loianno, K Daniilidis, and V Kumar. “Visual Servoing of Quadrotors for Perching by Hanging From Cylindrical Objects”. In: *IEEE Robotics and Automation Letters* 1.1 (Jan. 2016), pp. 57–64. DOI: 10.1109/LRA.2015.2506001. URL: http://ieeexplore.ieee.org/document/7347389/.

[24] T Hamel and R Mahony. “Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach”. In: *IEEE Transactions on Robotics and Automation* 18.2 (2002), pp. 187–198. DOI: 10.1109/TRA.2002.999647.

[25] R Mahony, P Corke, and T Hamel. “Dynamic Image-Based Visual Servo Control Using Centroid and Optic Flow Features”. In: *Journal of Dynamic Systems, Measurement, and Control* 130.1 (Dec. 2007). DOI: 10.1115/1.2807085. URL: https://doi.org/10.1115/1.2807085.

[26] B Herissé, T Hamel, R Mahony, and F Russotto. “Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow”. In: *IEEE Transactions on Robotics* 28.1 (2012), pp. 77–89. DOI: 10.1109/TRO.2011.2163435.

[27] P Serra, R Cunha, T Hamel, D Cabecinhas, and C Silvestre. “Landing of a Quadrotor on a Moving Target Using Dynamic Image-Based Visual Servo Control”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1524–1535. DOI: 10.1109/TRO.2016.2604495.

[28] N Franceschini, J.M. Pichon, and C Blanes. “From Insect Vision to Robot Vision”. In: *Philosophical Transactions of The Royal Society B Biological Sciences* 337 (Jan. 1992), pp. 283–294.

[29] D Coombs, M Herman, T Hong, and M Nashman. “Real-time obstacle avoidance using central flow divergence, and peripheral flow”. In: *IEEE Transactions on Robotics and Automation* 14.1 (1998), pp. 49–59. DOI: 10.1109/70.660840.

[30] J Conroy, G Gremillion, B Ranganathan, and J Humbert. “Implementation of wide-field integration of optic flow for autonomous quadrotor navigation”. In: *Autonomous Robots* 27 (Oct. 2009), pp. 189–198. DOI: 10.1007/s10514-009-9140-0.

[31] J Chahl, M.V. Srinivasan, and S Zhang. “Landing Strategies in Honeybees and Applications to Uninhabited Airborne Vehicles”. In: *I. J. Robotic Res.* 23 (Feb. 2004), pp. 101–110. DOI: 10.1177/0278364904041320.

[32] F Kendoul. “Four-dimensional guidance and control of movement using time-to-contact: Application to automated docking and landing of unmanned rotorcraft systems”. In: *The International Journal of Robotics Research* 33.2 (2014), pp. 237–267. DOI: 10.1177/0278364913509496. URL: https://doi.org/10.1177/0278364913509496.

[33] M Egelhaaf, N Böddeker, R Kern, R Kurtz, and J Lindemann. “Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action”. In: *Frontiers in Neural Circuits* 6 (2012). DOI: 10.3389/fncir.2012.00108. URL: https://www.frontiersin.org/articles/10.3389/fncir.2012.00108.

[34] T.S. Collett. “Insect Vision: Controlling Actions through Optic Flow”. In: *Current Biology* 12.18 (2002), R615–R617. DOI: https://doi.org/10.1016/S0960-9822(02)01132-6. URL: https://www.sciencedirect.com/science/article/pii/S0960982202011326.

[35] S Ravi et al. “Bumblebees display characteristics of active vision during robust obstacle avoidance flight”. In: *Journal of Experimental Biology* 225.4 (Feb. 2022), jeb243021. DOI: 10.1242/jeb.243021. URL: https://doi.org/10.1242/jeb.243021.

[36] N Sanket, C Singh, K Ganguly, C Fermüller, and Y Aloimonos. *GapFlyt: Active Vision Based Minimalist Structure-Less Gap Detection For Quadrotor Flight*. Vol. PP. Feb. 2018. DOI: 10.1109/LRA.2018.2843445.

[37] O.J.N. Bertrand, J.P. Lindemann, and M Egelhaaf. “A Bio-inspired Collision Avoidance Model Based on Spatial Information Derived from Motion Detectors Leads to Common Routes”. In: *PLOS Computational Biology* 11.11 (Nov. 2015), e1004339–. URL: https://doi.org/10.1371/journal.pcbi.1004339.

[38] H.W. Ho, G.C.H.E. De Croon, E van Kampen, Q.P. Chu, and M Mulder. “Adaptive Gain Control Strategy for Constant Optical Flow Divergence Landing”. In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 508–516. DOI: 10.1109/TRO.2018.2817418.

[39] J.J. Hagenaars, F Paredes-Vallés, Sander M Bohté, and G.C.H.E. de Croon. “Evolved Neuromorphic Control for High Speed Divergence-based Landings of MAVs”. In: *CoRR* abs/2003.03118 (2020). URL: https://arxiv.org/abs/2003.03118.

[40] J Dupeyroux, J.J. Hagenaars, F Paredes-Vallés, and G.C.H.E. De Croon. “Neuromorphic control for optic-flow-based landings of MAVs using the Loihi processor”. In: *CoRR* abs/2011.00534 (2020). URL: https://arxiv.org/abs/2011.00534.

[41] S Doncieux, N Bredeche, J Mouret, and A.E. Eiben. “Evolutionary Robotics: What, Why, and Where to”. In: *Frontiers in Robotics and AI* 2 (2015). DOI: 10.3389/frobt.2015.00004. URL: https://www.frontiersin.org/articles/10.3389/frobt.2015.00004.

[42] M Keshmiri, W -F. Xie, and A Mohebbi. “Augmented Image-Based Visual Servoing of a Manipulator Using Acceleration Command”. In: *IEEE Transactions on Industrial Electronics* 61.10 (2014), pp. 5444–5452. DOI: 10.1109/TIE.2014.2300048.

[43] W Zhao, H Liu, and X Wang. “Robust visual servoing control for quadrotors landing on a moving target”. In: *Journal of the Franklin Institute* 358.4 (2021), pp. 2301–2319. DOI: https://doi.org/10.1016/j.jfranklin.2021.01.008. URL: https://www.sciencedirect.com/science/article/pii/S0016003221000223.

[44] G.L. Mariottini, G Oriolo, and D Prattichizzo. “Image-Based Visual Servoing for Nonholonomic Mobile Robots Using Epipolar Geometry”. In: *IEEE Transactions on Robotics* 23.1 (2007), pp. 87–100. DOI: 10.1109/TRO.2006.886842.

[45] C McCarthy, N Barnes, and R Mahony. “A Robust Docking Strategy for a Mobile Robot Using Flow Field Divergence”. In: *IEEE Transactions on Robotics* 24.4 (2008), pp. 832–842. DOI: 10.1109/TRO.2008.926871.

[46] F Van Breugel, K Morgansen, and M.H. Dickinson. “Monocular distance estimation from optic flow during active landing maneuvers”. In: *Bioinspiration & biomimetics* 9.2 (2014), p. 025002.

[47] H.W. Ho and Y Zhou. “Incremental Nonlinear Dynamic Inversion based Optical Flow Control for Flying Robots: An Efficient Data-driven Approach”. In: *arXiv e-prints* (July 2023), arXiv:2307.02702. DOI: 10.48550/arXiv.2307.02702.

[48] E.J.J. Smeur, Q Chu, and G.C.H.E. de Croon. “Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles”. In: *Journal of Guidance, Control, and Dynamics* 39.3 (Dec. 2015), pp. 450–461. DOI: 10.2514/1.G001490. URL: https://doi.org/10.2514/1.G001490.