

Insect-Inspired Visual Perception for Flight Control and Collision Avoidance

Thèse N°9316

Présentée le 14 juin 2019

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de systèmes intelligents
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Julien Jean Denis Nicolas LECOEUR

Acceptée sur proposition du jury

Dr J. Skaloud, président du jury
Prof. D. Floreano, Dr. E. Baird, directeurs de thèse
Prof. H. Krapp, rapporteur
Prof. S. Viollet, rapporteur
Prof. P. Ramdya, rapporteur

2019

*“they are not as intelligent as we who kill them;
although they are more noble and more able.”*

—
Ernest Hemingway, *Old Man and the Sea*

To my son...

Acknowledgements

First and foremost, I would like to thank my two supervisors, Prof. Dario Floreano and Prof. Emily Baird for their trust, support and guidance during my PhD. Dario provided me with an active and stimulating research environment. He gave me a lot of freedom in my project, and taught me how to best present my work. I am grateful to Emily for her constant flow of ideas and always supportive feedback. She triggered my interest for insect behaviour modelling by hosting me in the Lund Vision Group, which has been a wonderful experience. Working at the boundary between robotics and biology is a fine art of balance and I am thankful to Dario and Emily for making this possible.

I would like to thank the members of my jury: Dr. Jan Skaloud, Prof. Pavan Ramdya, Prof. Holger Krapp and Prof. Stéphane Viollet for taking the time to read my thesis and for travelling to Lausanne for my thesis defense.

I am grateful to EPFL for providing a great studying environment, to the Swiss National Science Foundation and The Swedish Foundation for Strategic Research for funding this research, to Corinne Lebet for her work at the EDRS doctoral school, and to Vivek Ramachandran for being a caring student representative.

Thanks to Felix Schill, for starting what became the MAV'RIC autopilot. This project was for me a fantastic opportunity to discover many aspects of autopilot design (from electronics, to sensor drivers, communication, sensor fusion, and flight control) that I would have never dared exploring without an initial push from Felix. Although this project was discontinued in the lab, I am glad that it contributed to several research projects and publications, and I am thankful to the people who contributed: Nicolas Dousse, Grégoire Heitz, Felix Schill, Géraud L'Éplatténier, Basil Huber, Matthew Douglas, Alexandre Cherpillot, Jean-François Burnier, Dylan Bourgeois, Ludovic Daler, Anand Baskaran, Pu Bai and Meysam Basiri. Thanks to Przemyslaw Kornatowski and Ludovic Daler for their help in mechanical design. Thanks to Géraud L'Éplatténier for his patience with the CurvACE prototypes.

During my years in LIS, I supervised 33 students during their semester or master projects.

Acknowledgements

They all contributed to this thesis in one way or another and I am grateful to all of them for their hard work. Supervising students, as well as preparing and lecturing classes, came as a humble confirmation for what my former maths teacher François Lachaux once told me: you never truly understand something until you can teach it.

I would like to express my gratitude to the wonderful colleagues and friends with whom I spent time outside the lab at the Lund Vision Group: Atticus, Aravin, Basil, Gavin, John, Lana, Mindaugus, Nele, Nellie, Olivier, Therese, and others, as well as at the Laboratory of Intelligent Systems: Adrien, Alice, Alice (the other one), Basil (aka Alex), Fabian, Géraud, Grégoire, Ludovic, Nicolas, Olex, Przemek, Sebastian, Stefano, Vivek, and many others.

I also have a special thought for the bumblebees who contributed to this thesis in their own way. They are lovely creatures who deserve respect. I did my best to give them good living conditions with a peaceful environment and tasty flowers, and I do not hold any resentment against the one angry bumblebee who stung my finger.

I reserve my final thanks to my loving family, and specially to Lucie for her love, for bearing with me during difficult times, for carrying our child, and for making it all worthwhile.

Lausanne, 03 December 2018

J. L.

Abstract

FLYING ROBOTS are increasingly used for tasks such as aerial mapping, fast exploration, video footage and monitoring of buildings. However, autonomous missions are usually constrained to high altitude flights above flat surfaces in order to avoid collisions. Autonomous flight at low altitude in cluttered and unknown environments is an active research topic because it poses challenging perception and control problems. Traditional methods for collision-free navigation at low altitude require heavy resources to deal with the complexity of natural environments, something that limits the autonomy and the payload of flying robots.

Flying insects, however, are able to navigate safely and efficiently using vision as the main sensory modality. Flying insects rely on low resolution, high refresh rate, and wide-angle compound eyes to extract angular image motion – optic flow – and move in unstructured environments, rather than on overlapping, single-lens eyes and depth-from-stereo vision as many other animals do. These strategies result in systems that are physically and computationally lighter than those often found in high-definition stereovision.

Taking inspiration from insects offers great potential for building small flying robots capable of navigating in cluttered environments using lightweight vision sensors. Robots can be programmed to adopt insects strategies, but little is known about the way insects turn visual stimuli into muscular actuation during flight, especially in the presence of obstacles. This precludes a direct implementation of insect flight control into robots.

In this thesis, we investigate insect perception of visual motion and insect vision based flight control in cluttered environments. We use the knowledge gained through the modelling of neural circuits and behavioural experiments to develop flying robots with insect-inspired control strategies for goal-oriented navigation in complex environments.

We start by exploring insect perception of visual motion. We present a study that reconciles an apparent contradiction in the literature for insect visual control: current models developed to explain insect flight behaviour rely on the measurement of optic flow, however the most prominent neural model for visual motion extraction (the Elementary Motion Detector, or

Abstract

EMD) does not measure optic flow. We propose a model for unbiased optic flow estimation that relies on comparing the output of multiple EMDs pointed in varying viewing directions. Our model is of interest of both engineers and biologists because it is computationally more efficient than other optic flow estimation algorithms, and because it represents a biologically plausible model for optic flow extraction in insect neural systems.

We then focus on insect flight control strategies in the presence of obstacles. We perform experiments with bumblebees (*Bombus terrestris*) flying freely in specially designed environments in order to model their control strategy and study how it is affected by different methods for pooling visual motion across a panoramic field of view. By recording the trajectories of bumblebees, and by comparing them to simulated flights, we show that bumblebees rely primarily on the frontal part of their field of view, and that they pool optic flow in two different manners for the control of flight speed and of lateral position. For the control of lateral position, our results suggest that bumblebees selectively react to the portions of the visual field where optic flow is the highest, which correspond to the closest obstacles.

Finally, we tackle goal-oriented navigation with a novel algorithm that combines aspects of insect perception and flight control presented in this thesis – like the detection of fastest moving objects in the frontal visual field – with other aspects of insect flight known from the literature such as saccadic flight pattern. We present the design of a lightweight flying robot capable of producing lateral accelerations while limiting body rotations in order to facilitate optic flow measurement during insect-like saccadic flights. Through simulations, we demonstrate autonomous navigation in forest-like environments using only local optic flow information and assuming knowledge about the direction to the navigation goal.

Keywords: Flying robots, flying insects, optic flow, visual motion sensing, elementary motion detector, flight control, collision avoidance

Résumé

LES ROBOTS VOLANTS sont utilisés de plus en plus fréquemment pour des tâches telles que la cartographie aérienne, l'exploration rapide de zone difficiles d'accès, la prise de vue cinématographique ou encore l'inspection de bâtiments. Néanmoins, les vols automatiques sont généralement planifiés de façon à maintenir une altitude confortable au dessus du sol afin de réduire les risques de collision. Le vol autonome à basse altitude dans des environnements non cartographiés et encombrés par des obstacles est un sujet de recherche actif parce qu'il représente un réel challenge tant en terme de perception que de contrôle du vol. Les méthodes traditionnelles d'évitement de collision et de navigation à basse altitude nécessitent de lourds équipements embarqués ainsi qu'une forte puissance de calcul, deux éléments qui limitent le temps de vol et la charge utile des robots volants.

Les insectes volants, malgré leur poids plume et leur petite taille, sont capables de naviguer efficacement et en toute sécurité en utilisant leur vision comme principale source d'information. Les hommes et les robots traditionnels ont recours à une paire d'yeux à simple lentille, ce qui les dote d'une très bonne résolution binoculaire pour estimer les distances, mais aussi limite leur champ de vision ainsi que la vitesse d'acquisition des images. Au contraire, les insectes volants voient le monde au travers de leurs yeux à facettes, dont la faible résolution est compensée par une haute fréquence de rafraichissement et un champ de vision panoramique. Les insectes basent le contrôle de leur vol sur l'extraction de la vitesse de défilement des images, appelé le flux optique. Cela permet d'obtenir des systèmes dont la masse et les besoins en puissance de calcul sont plus faibles que ceux rencontrés dans les systèmes de vision binoculaires à haute résolution.

S'inspirer des insectes représente une excellente opportunité pour la conception de robots volants équipés de capteurs visuels miniatures et capables de naviguer de façon autonome dans des environnements encombrés d'obstacles. Bien qu'un robot pourrait être programmé afin de répliquer les stratégies employées par les insectes, il subsiste cependant de nombreuses incertitudes concernant la façon dont les insectes volants convertissent l'afflux d'informations visuelles en commandes motrices lors de vols en présence d'obstacles. À ce jour, cela empêche

de bénéficier de méthodes utilisées par les insectes dans une implémentation robotique de contrôle du vol en environnement complexe.

Dans cette thèse, nous étudions la perception du mouvement chez les insectes volants ainsi que leur contrôle de vol basé sur la vision en présence d'obstacles. Nous utilisons les résultats obtenus par le biais de modélisations de circuits neuronaux et d'expériences comportementales afin de développer des robots volants qui utilisent des stratégies de contrôle inspirées par les insectes pour naviguer dans des environnements encombrés par des obstacles.

Nous commençons par explorer la perception du mouvement visuel chez les insectes. Nous présentons une étude qui réconcilie une apparente contradiction dans la littérature liée au contrôle visuel du vol des insectes : ce dernier est expliqué par des modèles basés sur la mesure du flux optique, or le principal modèle neuronal pour l'extraction du mouvement visuel (le Elementary Motion Detector, ou EMD) ne mesure pas le flux optique. Nous proposons un modèle pour l'estimation du flux optique qui se base sur la comparaison des signaux de sortie d'une série d'EMDs pointés vers différentes directions dans le champ visuel. Notre modèle est intéressant non seulement pour les ingénieurs mais aussi pour les biologistes car en plus d'être plus efficace en termes de temps de calcul comparé aux autres algorithmes d'estimation du flux optique, il représente une hypothèse biologiquement plausible pour l'estimation du flux optique dans le système nerveux des insectes.

Nous nous concentrons ensuite sur le contrôle du vol en présence d'obstacles. Grâce à des expériences menées avec des bourdons (*Bombus terrestris*) volant librement dans des environnements spécialement conçus, nous modélisons leurs stratégies de contrôle et étudions la façon dont elle est affectée par différentes méthodes d'aggrégation du flux optique mesuré dans leur champ de vision panoramique. En mesurant les trajectoires des bourdons, et en les comparant avec des simulations de vol, nous montrons que les bourdons utilisent principalement la partie frontale de leur champ de vision, et qu'il aggrègent le flux optique de deux façons différentes pour le contrôle de leur vitesse de vol et celui de leur position latérale. Pour le contrôle de la position latérale, nos résultats suggèrent que les bourdons réagissent de façon sélective aux portions du champ de vision où le flux optique est le plus fort, ce qui correspond aux obstacles les plus proches.

Enfin, nous nous intéressons à la navigation autonome pour laquelle nous proposons un algorithme qui combine plusieurs aspects de la perception et du contrôle chez les insectes présentés dans cette thèse, notamment la détection des objets se déplaçant le plus rapidement dans le champ de vision frontal, ainsi que d'autres aspects issus de la littérature tels que le vol par saccades. Nous présentons le design d'un robot volant miniature capable de produire des accélérations latérales tout en minimisant les rotations de la caméra afin de faciliter la

mesure du flux optique pendant des vols en saccade inspirés par les insectes. Par le biais de simulations, nous démontrons la navigation autonome dans un environnement de type forestier en utilisant une information de flux optique mesurée localement.

Mots clefs : Robots volants, insectes volants, flux optique, perception visuelle du mouvement, elementary motion detector, contrôle du vol, évitement de collision

Contents

Acknowledgements	v
Abstract (English/Français)	vii
Table of contents	xv
List of figures	xviii
List of tables	xix
1 Introduction	1
1.1 Motivation	2
1.2 State of the art	4
1.2.1 Optic flow estimation in insect	4
1.2.2 Field of view and sensor orientation in insects and robots	5
1.2.3 Insect-inspired autonomous flight in complex environments	8
1.3 General approach	9
1.4 Main contributions and organization of the thesis	10
2 Insect-inspired estimation of optic flow	13
2.1 Introduction	15
2.2 Model	16
2.2.1 Predicted steady-state EMD response	17
2.2.2 Values of parameters	22
2.3 Theoretical Results	23
2.3.1 EMD Response Across the Visual Field	23
2.3.2 Deviation of Maximum EMD Response Ψ as Estimation of Relative Nearness	27
2.3.3 Threshold for an unambiguous estimation of relative nearness	27
2.3.4 Robustness to single-frequency pattern	29
2.4 Simulation results	31
2.4.1 Control strategy for lateral position and forward velocity	31

Contents

2.4.2	Simulation environment	33
2.4.3	Simulation results	34
2.4.4	Robustness to non-planar scenes	37
2.4.5	Robustness to single-frequency patterns	37
2.5	Robotic implementation	39
2.5.1	Hardware setup	39
2.5.2	Flight results	41
2.6	Discussion	41
2.6.1	Biological relevance of η_{\min}	42
2.6.2	Dynamic adaptation of the visual system	44
2.6.3	Coping with EMD measurement noise	45
2.6.4	Bi-dimensional field of view	46
2.7	Conclusion	46
3	The role of optic flow pooling in insect visual flight control	49
3.1	Introduction	51
3.2	Materials and Methods	53
3.2.1	Experimental setting	53
3.2.2	Geometrical expression of translational optic flow	54
3.2.3	Optic flow pooling	55
3.2.4	Force field for flight control	56
3.2.5	Prediction method	57
3.2.6	Prediction error	59
3.3	Results and discussion	60
3.3.1	The effect of obstacle density on lateral position	60
3.3.2	The effect of obstacle density on flight speed	63
3.3.3	The effect of obstacle density on vertical position	64
3.3.4	Predicting the effect of obstacle density and optic flow pooling on flight control	64
3.3.5	Predicted lateral position	64
3.3.6	Predicted flight speed	69
3.3.7	Predicted vertical position	71
3.3.8	Combining multiple optic flow pooling methods for different aspects of flight control	72
3.4	Conclusion	73
4	Joint obstacle avoidance and visual navigation in cluttered environments	75
4.1	Introduction	77

4.2	Method	78
4.2.1	Safe paths in forest environments	78
4.2.2	Insect saccadic flight	80
4.2.3	Obstacle detection	81
4.2.4	Gap selection and saccade generation	81
4.3	Simulation Results	82
4.4	Hardware implementation	84
4.4.1	Design of a multirotor robot for saccadic flight	84
4.4.2	Velocity control with level attitude	85
4.4.3	Six degrees of freedom control allocation	87
4.4.4	Results and limitations	88
4.5	Conclusion	90
5	Concluding remarks	91
5.1	Main accomplishments	92
5.2	Future work and outlook	93
5.2.1	Combining deep learning methods with bio-inspired models	93
5.2.2	Behavioural study of insect goal-oriented flight in cluttered environments	95
A	Publications	97
B	MAV'RIC: Open source software library for the development of drones	99
	Bibliography	111
	Curriculum Vitae	

List of Figures

1.1	Artistic view of a bumblebee (<i>Bombus terrestris</i>) navigating through a forest. . .	3
1.2	Insect eye and optic flow.	5
1.3	Elementary Motion Detector.	6
1.4	Placement of optic flow sensors in flying insects and robots.	7
2.1	Geometry of the model	18
2.2	Apparent angular period for varying viewing direction and ratio between linear period and distance.	20
2.3	Distribution of EMD output R across the visual field for varying speed and varying distance	25
2.4	Comparison of raw EMD response and Ψ angle as estimators of relative nearness	26
2.5	Effect of eye resolution and EMD integration time on η_{\min} threshold and Ψ angle	28
2.6	EMD velocity response curve for a broadband signal and sinusoidal signal. . . .	29
2.7	EMD response and Ψ angle for single-frequency pattern.	30
2.8	Block diagram of the simulation.	32
2.9	Simulation environment.	33
2.10	EMD Response to Simulated Images.	35
2.11	State of simulated agent after convergence.	36
2.12	Simulated trajectories in non-planar scenes.	38
2.13	Simulated trajectory with single-frequency pattern.	38
2.14	EMD response and relative nearness estimation onboard fixed wing drone. . .	40
3.1	Experimental setting.	52
3.2	Illustration of the prediction method for maxf optic flow pooling.	58
3.3	Effect of obstacle density on flight control in bumblebees.	61
3.4	Predicted lateral position, flight speed and vertical position.	65
3.5	Prediction errors for lateral position, vertical position and speed.	66
3.6	Prediction error when combining multiple optic flow pooling methods, with coupled forward and vertical control.	67

List of Figures

3.7	Predicted lateral position, flight speed and vertical position with coupled forward and vertical control.	68
4.1	Illustration of the navigation algorithm	79
4.2	Navigation results in simulation	83
4.3	BumbleBot: A lightweight flying robot for optic flow based navigation.	85
4.4	Schematic view of the controller.	86
4.5	Sample trajectories recorded during hardware experiments	89
5.1	Experimental setup for bumblebee navigation in cluttered environment	94
B.1	Research platforms built using the MAV'RIC software library.	99



List of Tables

2.1	Parameters values used during simulations.	22
2.2	Apparent angular period and temporal frequency for extreme parameter values.	23
3.1	Summary of the experimental measurements and statistical significance of the differences measured between pairs of conditions.	62

1 Introduction

THE GOAL of this thesis is to contribute to the fields of flying insect vision and bio-inspired autonomous flying robotics. We aim at proposing models for the visual perception of flying insects and its application to flight control in cluttered environments in order to implement autonomous navigation strategies that require lightweight visual sensors and low computational power. In this introductory chapter, the motivation behind this work is presented, followed by the state of the art. The main contributions of this thesis are then presented and its organization outlined.



1.1 Motivation

Autonomous flying robots are becoming increasingly popular for tasks such as aerial cartography, monitoring of buildings, video footage, or even automated delivery of small packages [46, 29]. In spite of the ever-increasing reliability of commercially available products as well as hobbyist projects, autonomous flying robots are usually constrained to flights at relatively high altitude over flat terrains in order to avoid collisions with obstacles on the ground. Autonomous navigation in complex, unstructured and unknown environments such as urban, forest and indoor environments is an active field of research because it seemingly requires robots with incompatible attributes: low weight, agile flight, advanced sensing and fast computation.

When flying in the sky, away from obstacles, the flying robot can rely on GPS positioning in order to navigate between predefined waypoints. While GPS signal is effective at high altitude, it becomes unreliable when flying at low altitude, in cluttered environments or in urban canyons, where it suffers from occluded satellites and from signal reflection on buildings. A more effective approach that does not require the knowledge of absolute position in a map consists in continuously monitoring the 3D layout of the surrounding environments in order to detect the presence of obstacles and to steer around them. Traditionally, autonomous navigation in unknown environments requires a real-time mapping of the surroundings (with methods such as SLAM [112, 25, 69]) and associated path-planning algorithms [44] to guaranty that the robot does not collide with obstacles. This safety usually comes at the cost of increased complexity, weight and limited flight speed due to the time needed to update the internal map and to compute a collision-free path.

If we take a look at Nature, we can appreciate the smart and efficient flight of insects in complex natural environments. Flying insects possess limited computational resources [41] and eyes with low spatial resolution [68]. Yet they are able to use vision to navigate effortlessly in cluttered and unknown environments [40, 124]. To achieve this, flying insects extract the pattern of visual motion generated in their panoramic field of view as they move (known as optic flow [53, 64]). They use this visual information with computationally efficient and reactive strategies to control their speed [123, 4, 7, 62, 100, 1], avoid obstacles [122, 62, 79], control their height above the ground [99, 101, 119], and to land [21, 41, 132, 125, 132, 133].

These highly efficient vision-based flight control strategies have attracted considerable interest among the community of scientists working in autonomous robotics over the last decade. Roboticists successfully took inspiration from flying insects to implement computationally efficient and lightweight solutions for autonomous flight stabilisation [94, 92], altitude regula-



Figure 1.1 – Artistic view of a bumblebee (*Bombus terrestris*) navigating through a forest, dodging tree trunks and encountering conspecifics on its path.

tion [109, 15, 47], landing [130, 15], obstacle avoidance [138, 15, 63] and speed estimation [67, 24, 126].

To date, however, most of the biological studies on optic flow based control were performed within environments made of flat surfaces that are far from the topology of natural environments, so it is not clear if the principles described in the literature stand true in more complex environments. Insects were presented with cubic rooms [128, 132] or corridors [4, 122, 62, 117] with varying width and height, but little is known about which part of their visual field they use [7, 80], or whether they are able to selectively extract information about specific areas. Moreover, it is still unclear how insects measure optic flow because the most prominent neuronal model for motion sensing suffers from imperfections [135].

In this thesis, we aim to address these limitations and to enable autonomous flight in complex and unknown environments while refining our knowledge on insect flight control. For this purpose, we study insects flying in complex environments and model their behaviour with a formalism suitable for both animals and robots.

1.2 State of the art

1.2.1 Optic flow estimation in insect

Optic flow is defined as the visual motion perceived by an observer as it moves through a scene. It consists of a 2D vector on the image plane, and its magnitude depends on both the relative speed of motion between the observer and surrounding objects, and on the structure of the environment, i.e. the shape and layout of the surrounding objects and their distance to the observer. In static environments, optic flow solely depends on the self motion of the observer and its distance to obstacles, which makes it suitable for ego-motion estimation and collision avoidance.

When it is computed in several locations in the visual field, optic flow forms a 2D vectors field that should not be confused with the 2D vector field of image angular velocity (Fig. 1.3). While the optic flow field is computed from the time variation of pixel intensities, the image angular velocity can be derived geometrically from the position and speed of the observer and of the distance to objects [64]. The optic flow field is ideally equal to the image angular velocity field, but this is not always the case. Discrepancies may occur when the image does not have sufficient contrast, or when a narrow field of view combined with unidirectional features lead to the aperture problem.

The general formula for the optic flow in the ideal case is the following (adapted from [64]):

$$\vec{OF}(\vec{u}) = -\frac{\vec{V} - (\vec{V} \cdot \vec{u})\vec{u}}{D(\vec{u})} - \vec{\Omega} \times \vec{u} \quad (1.1)$$

where $\vec{OF}(\vec{u})$ is the vector of optic flow in the viewing direction defined by the unit vector \vec{u} ; \vec{V} and $\vec{\Omega}$ are the translation and rotation speed of the observer, respectively; and $D(\vec{u})$ is the distance to the object in the viewing direction \vec{u} . In the case of pure translational motion, this expression can be simplified as:

$$\|\vec{OF}_{\text{trans}}(\vec{u})\| = \frac{\|\vec{V}\|}{D(\vec{u})} \sin(\alpha) \quad (1.2)$$

where α is the angle between the direction of motion and the viewing direction \vec{u} . Translational optic flow is thus proportional to flight speed and inversely proportional to distance. In other words it is inversely proportional to the time to contact, and a high value of translational optic flow value indicates an imminent collision.

In the part of the insect neural system that is involved in visual processing, Elementary motion detectors (EMDs) are responsible for the extraction of image motion between neighbouring

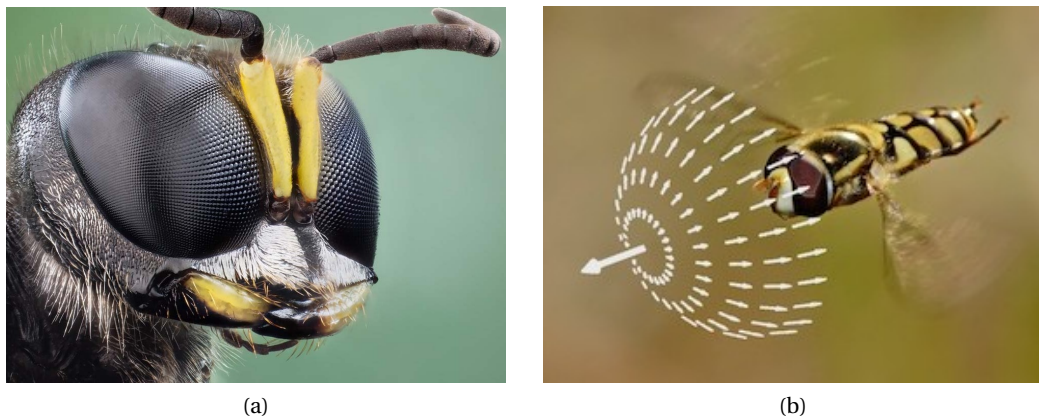


Figure 1.2 – Insect eye and optic flow.

(a): Insect compound eye. Each ommatidia composing an insect eye contains a miniature lens and has a relatively small field of view, but the compound eyes achieve a very wide field of view. (b): Illustration of optic flow field during translational flight. Note that its magnitude is null in the direction of flight, and that it increases towards the lateral, ventral and dorsal regions of the field of view.

ommatidia. Originally introduced by Hassenstein and Reichardt [56], this model is believed to be present in insect neural systems across the whole visual field and to be used in the early processing steps of the visual input [48, 89]. Their output is spatially integrated by a group of wide field motion sensitive neurons, each one applying a specific weighting function to its elementary motion inputs [49, 83, 67, 61]. These pooling neurons act like a series of motion sensors, each one with a different orientation and field of view. The Elementary Motion Detector, however, does not provide a perfect estimation of optic flow or image velocity. The EMD output is indeed highly correlated with image brightness and with the frequency content of the image [135].

1.2.2 Field of view and sensor orientation in insects and robots

Bumblebees regulate flight speed based on the image motion gathered in the lateral field of view, between 17° forward and 132° backward [7]. This result demonstrates that speed control is not performed based on information sensed in one unique direction, but rather based on the pooling of information coming from a large field of view. However, little is known regarding the relative importance of visual regions contained within these boundaries. In blowfly, flight speed is regulated by the fronto-lateral and moderately ventral eye region [62]. Optic flow vector fields were reconstructed in simulation from recorded flights trajectories. The visual areas with the lowest variation in optic flow amplitude during flights were designated as the

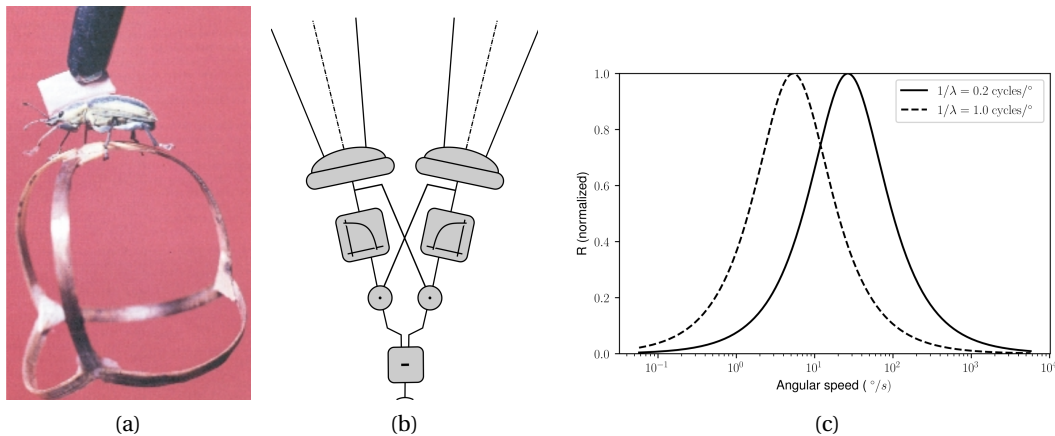


Figure 1.3 – Elementary Motion Detector.

(a): Original study by Hassenstein and Reichardt in 1956 which led to the EMD model [56, 17]. The beetle was observed turning left or right according to the presented visual stimulus. (b): Schematic view of the EMD model. It provides a motion estimate from two consecutive photoreceptors using only two low pass filters, two multiplications and a subtraction. (c): EMD response as function of the image angular speed. The response is maximum at a velocity that depends on the frequency of the visual stimulus. It thus cannot be used to measure angular velocity.

ones used for speed control. The limitation of such an approach is that no link was made between visual input and flight actuation output, nor any closed loop simulation performed, so nothing ensures that the fronto-lateral eye region was actively kept with a low variation of image motion by the animal. Indeed, in the hypothesis that suggests spatial pooling across the whole visual field, the average optic flow amplitude can be kept constant while local amplitudes are varying. It should instead be investigated whether there exists areas whose optic flow amplitude is highly correlated with changes in flight speed, denoting an active speed control.

In robotics, conflicting solutions for sensor placement can also be found. Optic flow sensors are often oriented at 90° to the side of the platform (for example [102]) because this is the orientation in which the optic flow amplitude is the highest when moving along a corridor. On the contrary, an orientation of 45° forward is a trade-off between the low optic flow amplitudes at angles near 0° and the already-past obstacles seen at angles above 90° (see [15]). Optic flow sensors oriented at 45° from the direction of motion also ensure that the measured optic flow is the maximum when flying toward a wall [139]. However, simulations with more complex environments found an optimum around 30° from the direction of motion [15].

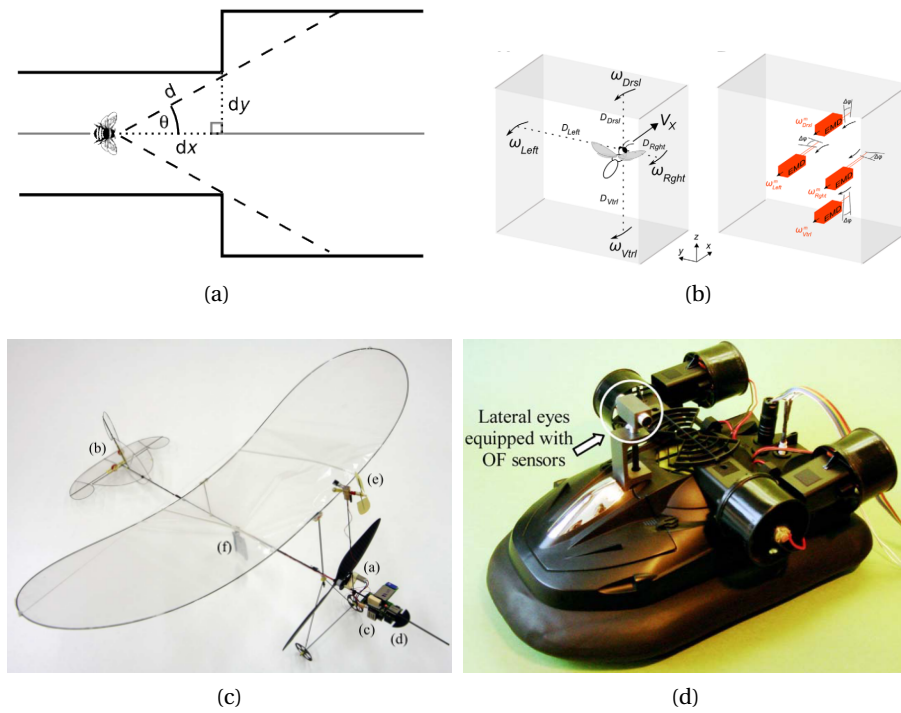


Figure 1.4 – Placement of optic flow sensors in flying insects and robots.
(a): Experiment showing that bumblebees use optic flow flexibly in the visual field, they react to a reduced corridor width by slowing down well before reaching the narrow section [7].
(b): Model on honeybee using two motion sensors oriented at 90° on the left and right sides and two motion sensors oriented at 90° upward and downward [98].
(c): Microflyer equipped with two lateral optic flow sensors oriented at 45° on the side, and one ventral optic flow sensor oriented at 45° downward [139].
(d): Hovercraft equipped with optic flow sensors oriented at 90° on the sides [116].

1.2.3 Insect-inspired autonomous flight in complex environments

A few attempts have been made to model insect flight control. The approach in [92, 93, 85, 33] is a bottom-up approach, which means that it attempts to build a simulated fly based on the current knowledge in insect neurology and aerodynamics. The main limitation of such method is that, in spite of the precision of neural recording on tethered flies, these results only apply to low level, reactive computations as the behaviour of the tethered animal is arguably different from that of a freely flying animal.

When the goal is to understand insect behaviour and to design a flying robot, rather than building or simulating a perfect copy of insect neural circuits and observing the resulting behaviour, an alternative method consists in proceeding the other way round: model animal behaviour and then infer and study the associated perception and control strategies (see for example [48, 41, 124, 119]).

The method presented in [22, 21] consists in representing each point of a flight trajectory as a vector in a space of dimension respectively six and four (three linear velocity plus respectively one and three angular velocities). Clustering algorithms are then used to identify prototypical motion patterns and objectively describe a trajectory. The same kind of approach is used in [52], the behaviour of the insect over time is modelled with a markov chain, each state corresponding to a prototypical motion. The transition probabilities between each state are estimated and the flight behaviour is predicted with fairly good accuracy. However, the probabilities to transition between states do not take into account the visual motion experienced by the insect, although insects are known to react according to the visual motion they perceive.

Vision-based autonomous indoor flight of a 30 grams microflyer was demonstrated using a control method inspired by insects [139]. Two controllers for altitude and lateral steering were used while flight speed was regulated with an anemometer. However, this demonstration was limited to a simple cubic environment with flat walls, ground and ceiling, and was highly dependant upon the texture used on the walls. Similarly, a fixed wing optic flow based controller, called OptiPilot, was developed for attitude stabilization, altitude regulation, obstacle avoidance, take-off and landing [15]. The controller consisted of two sets of weighted sums, for pitch and roll control, applied to the signal of seven optic flow sensors. The platform was able to avoid groups of trees in an open field. However, the platform was limited to flights at relatively high altitude at which the ground can be considered as a flat surface, and to large obstacles relative to the size of the robot). Results in simulation showed that it was not suited for small obstacle avoidance, and that it was subject to crashes in case of symmetric input, like when facing a wall. More sensors and a more refined strategy would be needed to detect smaller obstacles and deal with symmetric cases. It should also be noted that biological

studies tend to show that separate controllers for different tasks coexist in neural system of insects [121, 2]. Two cameras equipped with fish-eye lenses are used to extract optic flow from panoramic images in [127, 130]. The papers demonstrates vision-based stabilization and autonomous flight capabilities. Optic flow disparities are used in conjunction with usual image processing algorithms to detect moving objects. The presented methods, although effective, require flight at relatively high altitude over flat terrain. Finally, the resolution of the cameras and the multiple image processing tasks executed in parallel have a high computational cost and the method could not be miniaturized with the current performance of micro-controllers.

A ground robot controller derived from the optic flow sensitivity maps of lobula plate tangential cells [61] was shown to be robust to the presence of obstacles although it was not specifically designed to avoid obstacles. Environmental uncertainty was included in [63] to account for the unknown position of obstacles during the design of a quadcopter controller. This resulted in more robust tracking of an obstacle-symmetric trajectory, i.e. a trajectory with equal distance to the obstacles on the left and right sides. However this method does not allow to navigate towards a goal without breaking the obstacle symmetry of the trajectory. Combined navigation and obstacle avoidance was achieved with a simulated agent flying through a field of obstacles in [14]. The agent performs saccadic flight, i.e. a series a straight path segments and sharp yaw turns, similar to what is observed in flying insects. During the straight path segments, EMDs are used to gather relative nearness information around the agent and to take a decision regarding the amplitude of the next saccade. Saccades are performed in order to align the heading in a direction resulting from a weighted sum between the direction to the goal and the direction where obstacles are the closest on average around the agent. Because the proximity to obstacles is averaged all around the agent, however, there is no guaranty that this weighted sum will not result in the a saccade towards a small obstacle located in the direction opposite to that of average maximum proximity.

1.3 General approach

Throughout this thesis, we orient our research toward questions that are relevant for both biologists and roboticists. We concurrently explore flight control in complex environments through behavioural experiments with bumblebees and simulations and experiments on flying robots. Bumblebees (*Bombus terrestris*) were chosen as our model species because they are readily available and fly year round in the animal facilities of Lund University, and can be easily trained to fly in controlled environments between their hive and a food source. Finally, compared to smaller insects studied more extensively –like *Drosophila melanogaster*– bumblebees have a higher inertia, leading to flight dynamics closer to that of flying robots.

The simulation tools developed for this thesis keep an abstraction layer between real insects and simulated ones, as well as between real robots and simulated ones. Indeed, the first simulations compute idealized optic flow, and are thus not affected by optics quality, sensor resolution or image contrast. This way, experimental data can be analysed and modelled regardless of sensing modalities. For the final steps of the project, more realistic visualization is added. Although a significant part of this thesis is done through simulation, several robotic platforms were specifically designed to be capable of the manoeuvres observed in insects in order to ease the transfer of controllers from insects to robots.

1.4 Main contributions and organization of the thesis

This thesis contributes to both biology and robotics. We perform behavioural experiments on flying insects to model and understand their flight control strategies in cluttered environments. We perform computer simulations of autonomous agents in order to validate our biological models and to present them in a readily implementable fashion. We design and test flying robots in order to assess the applicability of the proposed strategies on real hardware.

The main novelty of this thesis is the use of the location of fastest image motion for perception and control. The presented work is articulated around the notion that locating a maximum response in a panoramic visual field provides additional information about the environment and about self motion, and that this information can be used to estimate the state of an agent or control its trajectory. This notion is applied to the estimation of optic flow, to the control of flight speed and lateral position, and to the navigation in the presence of obstacles.

The thesis is organized as follows:

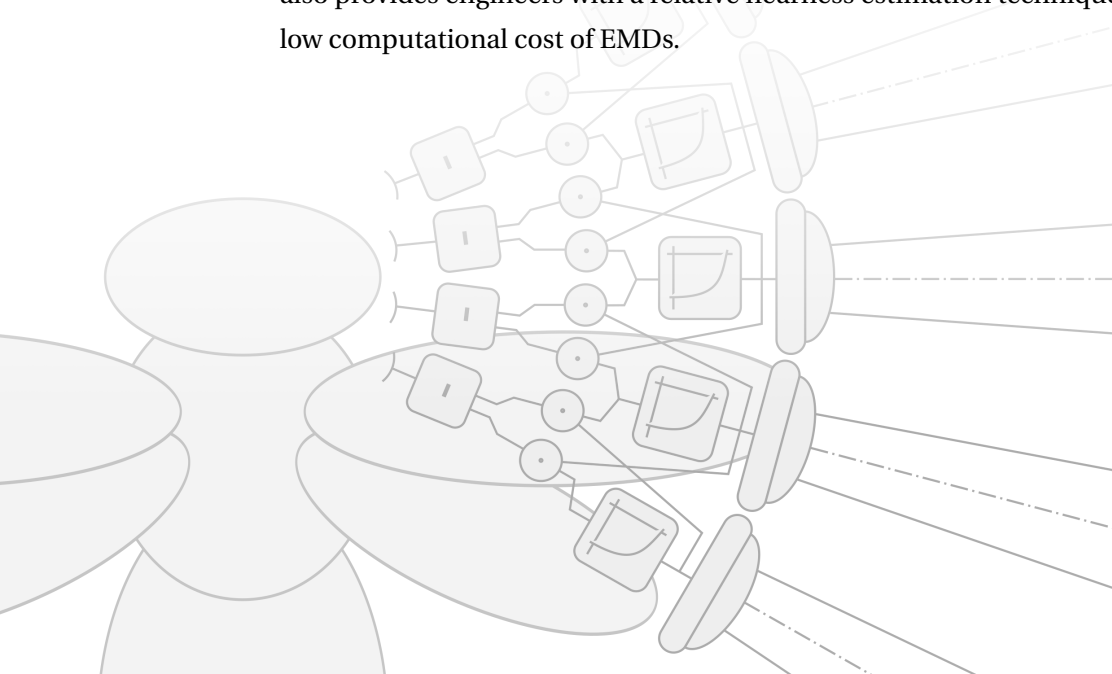
- In chapter 2, we tackle insect-inspired estimation of optic flow. We present a study on Elementary Motion Detectors (EMDs) and propose a novel method for the estimation of optic flow using a series of EMDs. We show that optic flow is spatially encoded in the visual field of flying insects by the location of the maximum EMD response. By relying on the location of maximum EMD response – rather than on the numeric value of this response – we propose an optic flow estimation method that does not suffer from the speed tuning of the EMD, nor from its dependency to image contrast. The method is studied through theoretical results, tested in simulation, and is implemented on a flying robot.
- In chapter 3, we tackle insect control of flight speed and lateral position in the presence of obstacles. We present the behavioural study and modelling of bumblebee flight in

the presence of obstacles. The study focuses on the effect of the pooling function used to combine optic flow measured across a panoramic visual field into control commands. We demonstrate that, to control their lateral position, bumblebees selectively react to the closest obstacles in front of them by locating the maximum image motion in the frontal visual field. This departs from existing models that either consider that optic flow is sampled at fixed locations in the visual field, or pooled through fixed weighted sums.

- In chapter 4, we tackle insect-inspired navigation in cluttered environments. We present an algorithm for joint navigation and obstacle avoidance in a forest-like environment. Our method uses the location of the fastest moving objects around a flying agent in order to generate yaw saccades towards a flight direction that is free of obstacles and that leads to the desired navigation goal. The method is tested in simulation, and we present the design of a flying robot that allows saccadic flight without parasitic body rotation.
- In chapter 5, we summarize and conclude the thesis before discussing potential improvements and opportunities for future work.

2 Insect-inspired estimation of optic flow

ELEMENTARY MOTION DETECTORS (EMD) are well-established models of visual motion estimation in insects. The response of EMDs are tuned to specific temporal and spatial frequencies of the input stimuli, which matches the behavioural response of insects to wide-field image rotation, called the optomotor response. However, other behaviours, such as speed and position control, cannot be fully accounted for by EMDs because these behaviours are largely unaffected by image properties and appear to be controlled by the ratio between the flight speed and the distance to an object, defined here as relative nearness. We present a method that resolves this inconsistency by extracting an unambiguous estimate of relative nearness from the output of an EMD array. Our method is suitable for estimation of relative nearness in planar scenes such as when flying above the ground or beside large flat objects. We demonstrate closed loop control of the lateral position and forward velocity of a simulated agent flying in a corridor. This finding may explain how insects can measure relative nearness and control their flight despite the frequency tuning of EMDs. Our method also provides engineers with a relative nearness estimation technique that benefits from the low computational cost of EMDs.



This chapter is adapted from the journal publication:

- [72] Lecoer, J., Baird, E. & Floreano, D. Spatial Encoding of Translational Optic Flow in Planar Scenes by Elementary Motion Detector Arrays. *Scientific Reports* **8**, 5821 (2018).

2.1 Introduction

Flying insects like flies, bees, moths and dragonflies are well known for their exquisite flight control capabilities. Despite their tiny brains and relatively crude visual systems, they routinely fly through cluttered environments, navigating over large distances and deftly avoiding obstacles in their path. To control their flight, insects use optic flow, defined as the pattern of apparent motion generated on their retina as they move through a scene [53]. Granted sufficient image texture, optic flow measures the apparent angular velocity of surrounding objects. For purely translational motion, translational optic flow (*TOF*) becomes proportional to the relative nearness — noted η — defined here as the ratio between the flight speed and the distance to an object [64]. Many complex behaviours exhibited by flying insects, such as visual odometry, landing, position, speed and height control are regulated using information extracted from optic flow (for reviews see [123, 124, 40]). Similar optic flow based strategies have also been successfully used to generate autonomous behaviour in miniature flying robots [15, 24, 110, 97, 45, 43], and even bio-hybrid robots [60]. Optic flow based strategies are interesting for the development of control systems in miniature flying vehicles because they have low computational cost and can be implemented on small platforms where constraints in weight and computational power are important.

The Elementary Motion Detector (EMD), introduced by Hassenstein and Reichardt [56, 9], is a well-established biological model for visual motion estimation. The model was originally developed to account for the turning responses made by walking beetles – known as the optomotor response – when presented with wide field yaw image motion [56] and has since been shown to match the optomotor responses of a wide range of insects [18]. The EMD performs spatio-temporal correlation of the signals from two adjacent photoreceptors and requires only two low-pass filters, two subtractions and one multiplication to provide an estimate of visual motion. This organisation is thought to exist in the early processing stages of the insect visual system in the form of hundreds of EMD units, each taking input from neighbouring photoreceptors around the panoramic field of view of insect eyes.

Neurophysiological studies [42, 3, 89, 50, 58, 66] have provided good evidence for the EMD as a candidate model for motion detection in insect brains, although recent literature shows evidence for both Barlow-Levick [10] and Hassenstein-Reichardt models [56, 9], suggesting a hybrid implementation (for reviews see [18, 19]). Indeed, models integrating the output of EMD arrays from a wide field of view – mimicking the tangential cells in the lobula plate of flies [89] – have been shown to detect the direction and amplitude of ego-rotations [20, 97], and to perform control of translational motion with simulated agents [94, 83, 33, 84, 14] and robotic agents [48, 105, 110, 118].

One of the key features of the EMD model is its dependency on the spatial structure of the scene [135, 38, 39, 11, 36]. Both the angular image speed tuning and the temporal frequency tuning of the EMD form a bell shape, with a maximum response at a frequency defined by its characteristics – namely, its integration time and interommatidial angle. While the frequency tuning of the EMD model mimics that observed in the optomotor response to rotational motion, strong support for the model as a basis for translational motion detection is lacking. Behavioural experiments suggest that insects are able to use translational optic flow to correctly estimate relative nearness independently of the spatial structure of the visual input [123, 125, 4, 81]. This is something that cannot be derived unambiguously from the raw EMD signals because of its bell-shaped tuning to angular speed that is not a monotonic function. Limitations in EMD-based control of translational motion due to the drop in EMD response at low distance from a surface, causing collisions into the surface, have also been reported [84, 14].

Here, we present a novel approach that suggests how the output of EMD arrays could indeed provide the basis for translational motion control in both insects and robotic agents. We show that, although the response of a single EMD does not provide a reliable measurement of angular image speed, comparing responses across an array of EMDs can provide an unambiguous estimate of relative nearness. We study analytically the response of an azimuthally distributed array of EMDs when moving along an planar surface covered by a pattern with a natural distribution of spatial frequencies [131, 8, 137, 115]. This surface models either large objects on the sides of the agent, or the ground below the agent. We show that, when the ratio between the speed of the agent and its distance to the surface is higher than a threshold we call η_{\min} , the angular location of the EMD with maximum response provides an unambiguous estimate of relative nearness. Our estimator performs best at low distance from the surface – in cases where the raw EMD output provides ambiguous estimates of relative nearness. We then discuss how this finding could be used for flight control, and how the model parameters could be dynamically adapted to enhance the relative nearness estimation. Finally, the proposed EMD-based relative nearness estimator is validated in closed-loop control of a simulated agent.

2.2 Model

Let us consider an agent — be it biological or artificial — flying in an environment composed of a flat surface (Fig. 2.1a). This surface could represent the ground below a flying agent, or one of the two vertical walls of the corridors commonly used for behavioural studies of insects and birds (for example [123, 117, 16, 80, 81, 7]).

The flying agent moves at speed V and distance d to the surface (Fig. 2.1a). Let us define the azimuth angle Φ as the angle between the front of the agent and a viewing direction. We will refer to the viewing direction $\Phi = 90^\circ$ as the "lateral region" of the field of view, but this could equally be the "ventral region" of the field of view if the surface was below the agent.

In order to mimic the properties of a real world environment, the flat surface is covered with a pattern that contains a natural distribution of spatial frequencies [131, 8, 137, 114, 115], i.e. its power spectrum follows a distribution of frequencies in $1/f^2$ (Fig. 2.1-inset).

The eye of the flying agent is composed of a planar array of photoreceptors (Fig. 2.1b). This plane is orthogonal to the patterned surface and it contains the agent velocity vector \vec{V} . Each photoreceptor points to a different azimuth angle Φ and has an acceptance angle $\Delta\rho$. Consecutive photoreceptors are separated by an angle $\Delta\Phi$. The receptivity function of an photoreceptor is approximated by a Gaussian window centered on Φ with standard deviation σ as in previous studies [120, 83, 33, 134, 95]. The acceptance angle of a photoreceptor — noted $\Delta\rho$ — is defined as the full width at half maximum of the Gaussian window [68].

A series of EMDs [9, 38, 39] takes input from the photoreceptor array. The output R_{Φ_i} of the EMD circuit pointed at the direction Φ_i is given by the difference of the results of two multiplications (Fig. 2.1b). The first multiplication is the product of the low-pass filtered signal of the photoreceptor pointed at $\Phi_i - \frac{\Delta\Phi}{2}$ and the unfiltered signal of the photoreceptor pointed at $\Phi_i + \frac{\Delta\Phi}{2}$. The second multiplication is the product of the unfiltered signal from the photoreceptor pointed at $\Phi_i - \frac{\Delta\Phi}{2}$ and the low-pass filtered signal of the photoreceptor pointed at $\Phi_i + \frac{\Delta\Phi}{2}$.

2.2.1 Predicted steady-state EMD response

In this section we derive the expression of the EMD output value R as a function of five parameters: the azimuth angle Φ , the agent speed V , the distance between the agent and the surface d , the inter-ommatidial angle $\Delta\Phi$, and the time constant τ of the EMD low-pass filter blocks.

The EMD used in this study is a balanced correlator [9] composed of two linear low pass filters, one multiplication and one subtraction. The mean EMD response to a moving broadband image can be expressed as the sum of its responses to the individual sinusoidal components of the image, weighted by the power spectrum of the image [36]. For a pattern containing a naturalistic distribution of frequencies – i.e. a power spectrum in $1/f^2$ – the mean EMD

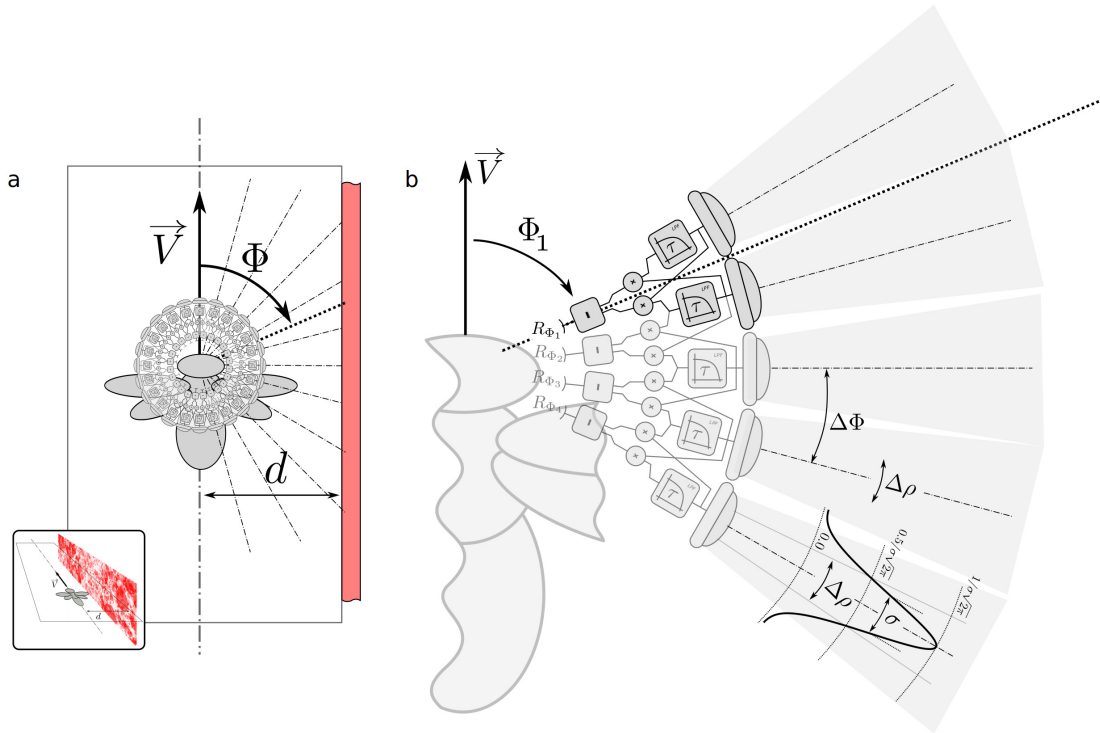


Figure 2.1 – Geometry of the model.

(a): Top view of the model. The flying agent — here represented as a bee — moves along a linear trajectory shown as a mixed dashed line. It flies at a speed V and at distance d from a flat surface, which is covered with a pattern that represents natural spatial frequencies. The agent sees the surface on its right. Viewing directions are defined by the angle Φ , with $\Phi = 0^\circ$ for the frontal viewing direction, $\Phi = 90^\circ$ for the viewing direction pointing to the right, and $\Phi = 180^\circ$ for the backward viewing direction. Overlaid on top of the agent is represented the array of photoreceptors and the array EMD networks considered in this study. The photoreceptors are aligned on a plane that is orthogonal to the patterned surface. (a-inset): Perspective view of the model. (b): Model of the eye and array of EMDs. The eye of the agent is composed of a planar array of independent photoreceptors here represented by five lens-like units. The network of EMDs is retinotopically organized with each EMD taking input from two consecutive photoreceptors. Each one of the four EMDs represented here is composed of two temporal low-pass filter blocks (square blocks labeled τ), two multiplication blocks (circular blocks labeled \times) and one subtraction block (square blocks labeled $-$).

response is thus given in equation (2.1).

$$R_{\Phi} = \int_{f_{min}}^{f_{max}} \frac{1}{f^2} R_{\Phi}^f df \quad (2.1)$$

where R_{Φ}^f is the response of the EMD pointed at the viewing direction Φ for a surface covered with a pattern that contains only one spatial frequency f , i.e. a sinusoidal grating. In equation (2.1) the integral computes summation across a range of frequencies. Note that this does not imply that frequency summation is implemented in insect nervous system and thus does not require additional neural computation. The frequency summation is however needed in this study to predict the EMD response to a signal that is itself the sum of sinusoidal components of varying frequencies.

The response R_{sin} of one EMD to a sinusoidal stimulus was derived in a previous study [39] for the case of a rotating drum patterned on its inner surface, and is shown in equation (2.2).

$$R_{sin} = \Delta I^2 \sin\left(2\pi \frac{\Delta\Phi}{\lambda}\right) \frac{\tau\omega}{1 + \tau^2\omega^2} \quad (2.2)$$

where ΔI is the amplitude of the sinusoidal stimulus, ω is the frequency of the stimulus, λ is its angular period, $\Delta\Phi$ is the inter-ommatidial angle, and τ is the time constant of the low pass filter.

While R_{sin} was derived with the assumption that ΔI , λ and ω were constant across the field of view [39], in our case (Fig. 2.1a) they vary depending on the azimuth angle as well as on the position and speed of the agent. Let us introduce the apparent signal amplitude $\widehat{\Delta I} = \Delta I(f, \Delta\Phi, \Phi, d)$, the apparent angular period $\widehat{\lambda} = \lambda(f, \Phi, d)$, and the apparent angular frequency $\widehat{\omega} = \omega(f, V)$. For example, the apparent angular period will decrease for increasing distance to the wall, the apparent angular period will also be maximum for $\Phi = 90^\circ$ and tend to 0 for $\Phi \rightarrow 0^\circ$ and $\Phi \rightarrow 180^\circ$. The expressions of $\widehat{\Delta I}$, $\widehat{\lambda}$, and $\widehat{\omega}$ are given below.

Apparent angular period The apparent angular period $\widehat{\lambda}$ is defined, for a given linear period Λ , as the angular size occupied by a complete cycle on the retina of the agent as shown on Fig. 2.2a. It is dependent on the distance to the surface d , the linear period Λ and the viewing direction Φ . By posing $x^+ = d \tan(\theta) + \frac{\Lambda}{2}$, $x^- = d \tan(\theta) - \frac{\Lambda}{2}$, and $\theta = \frac{\pi}{2} - \Phi$, we obtain geometrically from Fig. 2.2:

$$\widehat{\lambda} = \arctan\left(\frac{x^+}{d}\right) - \arctan\left(\frac{x^-}{d}\right) \quad (2.3)$$

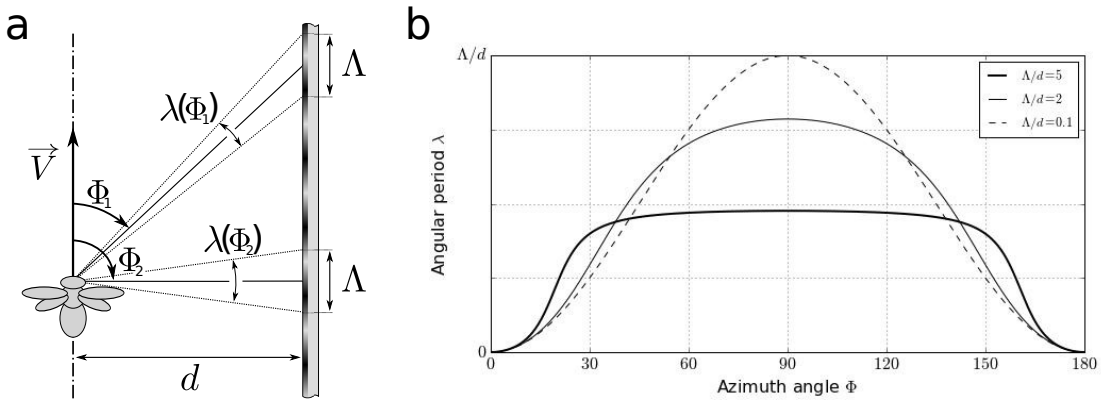


Figure 2.2 – Apparent angular period for varying viewing direction Φ and ratio between linear period Λ and distance d .

For $\Lambda \ll d$, the small angle approximation holds true and the apparent angular period follows a squared sinus as is expected in the case of a lateral infinite wall. However when $\Lambda > d$, the apparent angular period is shorter on the side ($\Phi \approx 90^\circ$), and longer in front and in the back ($\Phi \rightarrow 0^\circ$ and $\Phi \rightarrow 180^\circ$).

Thus the expression for the apparent angular period is given by equation (2.4). The variation of the apparent angular period across the visual field can be seen on Fig. 2.2b.

$$\hat{\lambda} = \arctan\left(\tan\left(\frac{\pi}{2} - \Phi\right) + \frac{\Lambda}{2d}\right) - \arctan\left(\tan\left(\frac{\pi}{2} - \Phi\right) - \frac{\Lambda}{2d}\right) \quad (2.4)$$

Apparent temporal and linear frequency Linear frequencies, linear periods and angular periods are respectively noted f , Λ and λ , and are respectively expressed in m^{-1} , m , and radians. The relation between f and Λ is given by equation (2.5).

$$f = \frac{1}{\Lambda} \quad (2.5)$$

Temporal frequency ω is expressed in radians per second. In the case presented in Fig. 2.2a, the temporal frequency does not depend on the viewing direction Φ and is only function of

the speed of the agent and the linear period as given in equation (2.6).

$$\hat{\omega} = 2\pi \frac{V}{\Lambda} \quad (2.6)$$

Apparent signal amplitude In a viewing direction Φ , a sinusoidal pattern is seen with an apparent angular period λ through photoreceptors with Gaussian acceptance windows of standard deviation $\sigma = \frac{\Delta\rho}{2\sqrt{2\ln(2)}}$. The apparent signal amplitude is the result of the convolution of the input signal with this Gaussian window as in equation (2.7).

$$\widehat{\Delta I} = \exp\left(-2\pi^2 \frac{\sigma^2}{\lambda^2}\right) \quad (2.7)$$

According to [68], in most diurnal insect species the ratio between acceptance angle and inter-ommatidial angle is given by equation (2.8):

$$\frac{\Delta\rho}{\Delta\Phi} \approx 1.07 \quad (2.8)$$

From equation (2.7) and equation (2.8), we can express the apparent signal amplitude as a function of the inter-ommatidial angle and the apparent angular period, as given in equation (2.9).

$$\widehat{\Delta I} \approx \exp\left(-2\pi^2 \frac{(0.45 \Delta\Phi)^2}{\lambda^2}\right) \quad (2.9)$$

Complete formulation and simplification We can thus reformulate equation (2.2) for our case as equation (2.10).

$$R_{\Phi}^f = \widehat{\Delta I}^2 \sin\left(2\pi \frac{\Delta\Phi}{\lambda}\right) \frac{\tau\hat{\omega}}{1 + \tau^2\hat{\omega}^2} \quad (2.10)$$

Chapter 2. Insect-inspired estimation of optic flow

By substituting equations (2.4, 2.5, 2.9) in equation (2.10), we obtain equation (2.11).

$$R_{\Phi}^f = \frac{2\tau\pi Vf}{1 + \tau^2(2\pi Vf)^2} \cdot \exp\left(\frac{-2\pi^2 (0.45\Delta\Phi)^2}{(A-B)^2}\right) \cdot \sin\left(\frac{2\pi\Delta\Phi}{A-B}\right) \quad (2.11)$$

$$, \text{ where } \begin{cases} A = \arctan(\tan(\frac{\pi}{2} - \Phi) + \frac{2\pi Vf}{2d}) \\ B = \arctan(\tan(\frac{\pi}{2} - \Phi) - \frac{2\pi Vf}{2d}) \end{cases}$$

The complete EMD output given by the integral in equation (2.1) and equation (2.11) is approximated as a discrete sum by considering a finite number $N_f \gg 1$ of spatial frequencies f_k , as shown in equation (2.12).

$$R_{\Phi} \approx \sum_{k=0}^{N_f-1} \frac{1}{f_k^2} R_{\Phi}^{f_k}, \text{ where } f_k = f_{min} + k \cdot \frac{f_{max} - f_{min}}{N_f - 1} \quad (2.12)$$

2.2.2 Values of parameters

The model was evaluated with $N_f = 2000$, $f_{min} = 1$, and $f_{max} = 1000$. The five parameters τ , $\Delta\Phi$, d , V , and Φ were linearly sampled according to the values presented in the following table.

	minimum	maximum	Number of values
f	1 m ⁻¹	1000 m ⁻¹	2000
τ	1 ms	10 ms	28
$\Delta\Phi$	0.5°	5°	28
d	0.05 m	0.2 m	49
V	0.1 m/s	0.8 m/s	49
Φ	$\pi/24$ rad	$\pi/2$ rad	200

Table 2.1 – Parameters values used during simulations.

The speed of the agent V was sampled with 49 different values between 0.1m/s and 0.8m/s. The distance d between the agent and the surface was sampled with 49 different values between 5cm and 20cm. The inter-ommatidial angle $\Delta\Phi$ was sampled with 28 different values between 0.5° and 5°. The time constant τ was sampled with 28 different values between 1 ms and 10 ms. The azimuth angle Φ was sampled with 200 different values between $\pi/24$ and $\pi/2$.

The range of spatial frequencies f_{\min} and f_{\max} was chosen so that they do not interfere with the results of the study. The maximum spatial period is several orders of magnitude larger than the length covered by the agent flying at the maximum speed during an EMD integration time (Table 2.2). The minimum spatial period is small compared to the length covered by one acceptance angle ρ at the smallest considered distance to the wall, and is thus filtered by the gaussian acceptance angle convolution, which also avoids potential issues of spatial aliasing [26].

λ_{90} (°)	d_{\min}	d_{\max}	$\nu = \frac{\omega}{2\pi}$ (cycles/s)	V_{\min}	V_{\max}
f_{\min}	168.58	136.40	f_{\min}	0.1	0.8
f_{\max}	1.15	0.29	f_{\max}	100.0	800.0

Table 2.2 – Apparent angular period and temporal frequency for extreme parameter values. **(left)**: Apparent angular period λ measured at $\Phi = 90^\circ$ for extreme values of distance d and linear frequency f . **(right)**: Temporal frequency ν at the maximum and minimum values of speed V and spatial frequency f .

2.3 Theoretical Results

In this section, we analyse theoretical predictions of the response of an EMD array to an planar surface covered with a natural pattern. We evaluated equation (2.12) for varying values of the five parameters Φ , V , d , $\Delta\Phi$ and τ (see Supplementary Table S1). These results are analysed in the following paragraphs.

We first show that the value of the EMD output is not a reliable estimation of relative nearness (i.e. V/d) in that a single value of EMD output can not be unambiguously associated to a single value of the V/d ratio. Then, we introduce the angle Ψ , which is obtained from the azimuthal location of maximum output in the EMD array. We show that the angle Ψ covaries monotonically – though non-linearly – with V/d , and thus provides an unambiguous estimate of relative nearness.

2.3.1 EMD Response Across the Visual Field

When a flying agent is moving in straight line at constant speed, in a purely translational motion (Fig. 2.1a), translational optic flow is proportional to flight speed V and inversely proportional to the distance to an object in the scene [64, 140]: $\text{TOF}(\Phi) = \frac{V}{D_\Phi} \sin(\Phi)$. For the planar surface shown in Fig. 2.1, which is aligned with the velocity vector and at a distance d from the agent, the distance to the surface in the viewing direction Φ is $D_\Phi = d/\sin(\Phi)$. Translational optic flow can then be obtained geometrically with equation (2.13) which is

positively correlated to flight speed V and inversely correlated to distance d . Note that translational optic flow at viewing angle $\Phi = 90^\circ$ yields a maximum value — noted TOF_{90} — that is equal to the relative nearness $\eta = V/d$.

$$TOF(\Phi) = \frac{V}{d} \sin^2(\Phi) \quad (2.13)$$

For a planar EMD array, the absolute value of the EMD response R increases at all azimuth angles with increasing flight speed in the range of flight speeds considered in our study (Fig. 2.3a). At higher flight speeds, the EMD response reaches a maximum then decreases with increasing flight speed (see Supplementary Figure S3). However, R does not always increase with decreasing distance to the surface (Fig. 2.3c), contrary to optic flow which increases with decreasing distance. For example, with the EMD parameters used for Fig. 2.3, R increases for decreasing values of d only in the extreme frontal and rear parts of the field of view (in the ranges $\Phi \in [0^\circ, 30^\circ]$ and $\Phi \in [150^\circ, 180^\circ]$). In most of the field of view ($\Phi \in [45^\circ, 135^\circ]$), R increases with increasing values of d , which is the opposite of a relative nearness estimator.

Let us define R_{90} as the EMD response at $\Phi = 90^\circ$, and R_{\max} as the maximum EMD response which is located at $\Phi = \Phi_{\max}$ (Fig. 2.3a). Neither R_{90} or R_{\max} provide a correct estimate of relative nearness. While they both depend on flight speed and distance to the surface (Fig. 2.4b and Fig. 2.4c), the isocurves of R_{90} and R_{\max} are not at a constant V/d ratio, as is the case for relative nearness (Fig. 2.4a). This means that, unlike relative nearness, a single V/d ratio can correspond to different R_{90} or R_{\max} values. An agent flying at speed V and distance d to the surface should measure the same relative nearness when flying at double speed and double distance because the ratio V/d is the same in both cases. However this is not the case for R_{90} and R_{\max} which yield two different values when the agent is flying at speed V at distance d , and at speed $2V$ at distance $2d$.

Conversely, a single value of R_{90} or R_{\max} can correspond to different V/d ratios. The ambiguity of the estimate provided by R_{90} and R_{\max} is clearly visible when they are displayed as function of V/d , i.e. the relative nearness (Fig. 2.4f and Fig. 2.4g). A single value of R_{90} or R_{\max} can correspond to a wide range of relative nearness. For instance, for $R_{90} = 0.06$ on Fig. 2.4f, the relative nearness can be anywhere between $2 \text{ rad}\cdot\text{s}^{-1}$ and $16 \text{ rad}\cdot\text{s}^{-1}$. Similarly, for $R_{\max} = 0.06$ on Fig. 2.4g, the relative nearness can be anywhere between $2 \text{ rad}\cdot\text{s}^{-1}$ and $10 \text{ rad}\cdot\text{s}^{-1}$.

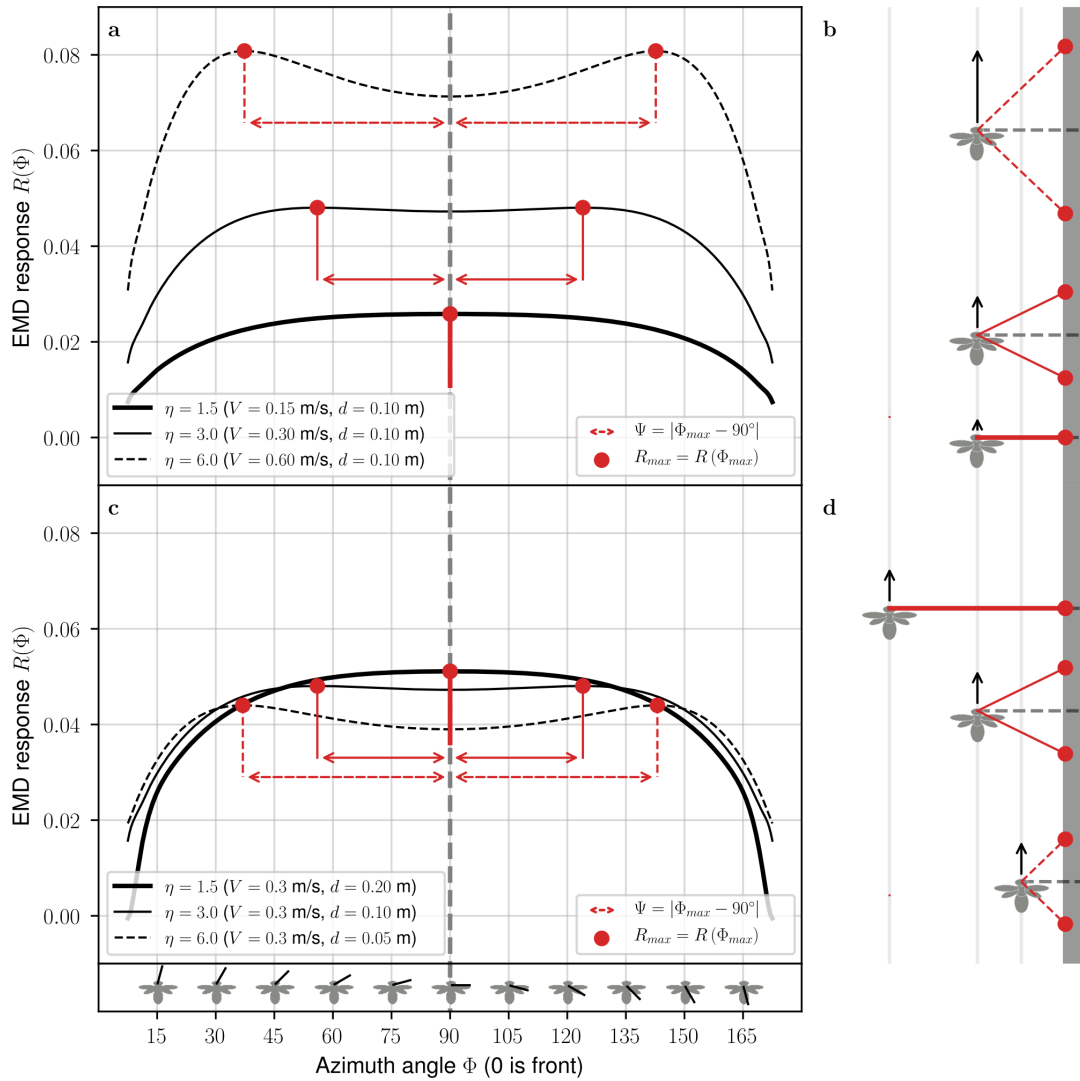


Figure 2.3 – Distribution of EMD output R across the visual field for varying speed and varying distance.

(a) and (c): The EMD output R is shown as a function of the azimuth angle Φ . Each black curve represents R for a specific value of V and d . The red dots represent, for each curve, the maximum EMD output across the visual field. The azimuth angle where R is maximum is noted Φ_{max} and the maximum value of R is noted R_{max} . The red arrows represent the angle Ψ (defined as $\Psi = |\Phi_{max} - 90^\circ|$) i.e. the angular deviation of the maximum EMD response from the side of the field of view ($\Phi = 90^\circ$). For both graphs, the inter-ommatidial angle and time constant of the low pass filter are kept constant at $\Delta\Phi = 3^\circ$ and $\tau = 10$ ms. (a) and (b): The distance to the surface d is kept constant at 10 cm for flight speeds 0.15 m/s, 0.30 m/s and 0.60 m/s. (c) and (d): The flight speed V is kept constant at 30 cm/s, for distances to surface of 5 cm, 10 cm and 20 cm. (b) and (d): Schematic representation of the agent flying alongside the vertical surface for the different values of V and d . The location of maximum EMD response is represented by red dots at the location they would project on the patterned surface. The angle Ψ is equal to 0 for the lower value of the ratio V/d (thick solid lines), and it increases with increasing V/d ratios (solid lines and dashed lines).

Chapter 2. Insect-inspired estimation of optic flow

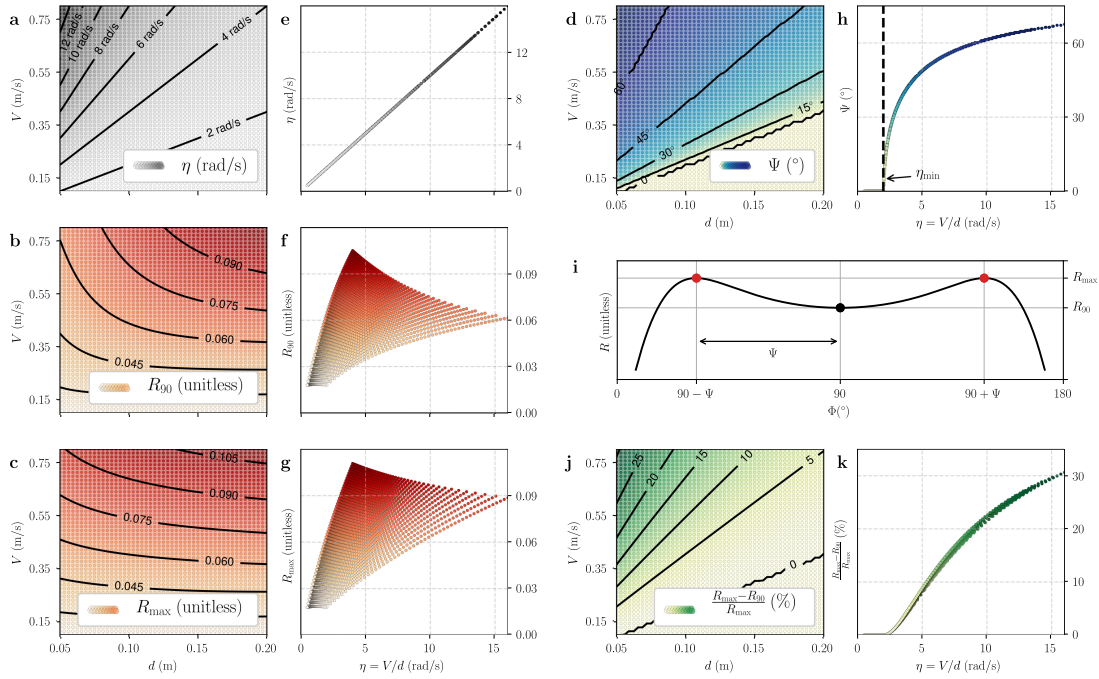


Figure 2.4 – Comparison of raw EMD response and Ψ angle as estimators of relative nearness. **(a)** and **(e)**: Relative nearness, computed geometrically as $\eta = V/d$. The unit is rad/s because this is equivalent to the angular image speed. **(b)** and **(f)**: The EMD response at 90° is defined as $R_{90} = R(\Phi=90^\circ)$. **(c)** and **(g)**: The maximum EMD response is defined as $R_{\max} = R(\Phi=\Phi_{\max})$. **(d)** and **(h)**: Deviation of the location of maximum EMD response $\Psi = |\Phi_{\max} - 90^\circ|$. Left **(a-d)**: Values given as functions of flight speed V and distance d . Right **(e-h)**: Values given as functions of the relative nearness η which is equivalent to the translational optic flow at viewing angle 90 degrees $\eta = TOF_{90} = V/d$. **(i)**: Graphical representation of R_{90} , R_{\max} and Ψ on EMD response R shown as function of viewing angle Φ . **(j-k)**: Relative difference between R_{\max} and R_{90} (given in percents), it indicates the maximum level of noise allowing the two maxima to be detected. In all plots, the inter-ommatidial angle and time constant of the low pass filter are kept constant at $\Delta\Phi = 3^\circ$ and $\tau = 10$ ms.

2.3.2 Deviation of Maximum EMD Response Ψ as Estimation of Relative Nearness

It is interesting to note that the maximum EMD response (noted R_{\max} and indicated by red dots in Fig. 2.3) is not always located where the translational optic flow (defined in equation 2.13 as the image angular velocity) is the highest, ie. at $\Phi = 90^\circ$. The location of the maximum EMD response (noted Φ_{\max}) is thus not equivalent to the location of the maximum translational optic flow. Let us define Ψ , the deviation of the maximum EMD response from the side of the field of view as

$$\Psi = |\Phi_{\max} - 90^\circ| \quad (2.14)$$

The fact that EMD response is not highest at $\Phi = 90^\circ$ can be explained by two facts. First, the bell-shaped speed tuning of EMDs when presented to broadband images [36] has a maximum at a specific angular speed (see Supplementary Figure S3). Second, the apparent image speed is lower in the frontal and rear parts of the visual field than at $\Phi = 90^\circ$ as shown in equation (2.13). Thus, at high relative nearness, the EMD may respond with a larger value to the lower angular image speed at $\Phi = 90 \pm \Psi$, than to the larger angular image speed at $\Phi = 90^\circ$.

With fixed distance to the surface, Ψ increases with increasing flight speed (Fig. 2.3a and Fig. 2.3b). With fixed flight speed, Ψ increases with decreasing distance (Fig. 2.3c and Fig. 2.3d). Thus, Ψ is increasing for increasing values of the ratio V/d , which is the relative nearness. As a consequence, we propose to use Ψ — rather than R — to estimate relative nearness.

Contrary to R_{90} and R_{\max} , the isocurves of Ψ are at a constant V/d ratio (Fig. 2.4d), which is also the case for the relative nearness (Fig. 2.4a). Moreover a single value of Ψ corresponds to a single V/d ratio (Fig. 2.4h), like relative nearness (Fig. 2.4e).

The function $\eta \mapsto \Psi$ is monotonically increasing (Fig. 2.4h). However this function is not strictly increasing for the lower values of η where $\Psi = 0^\circ$ (left region of Fig. 2.4h). This means that Ψ can be used to compare relative nearness in different regions of the field of view (as described later in the experimental section) only when relative nearness is higher than a threshold value.

2.3.3 Threshold for an unambiguous estimation of relative nearness

The deviation of maximum EMD response Ψ is equal to zero (i.e. $\Phi_{\max} = 90^\circ$) for all values of η below a threshold η_{\min} (lower right corner of Fig. 2.4d and left region of Fig. 2.4h). If $\eta < \eta_{\min}$, then Ψ is null and provides no useful information on relative nearness. However if $\eta > \eta_{\min}$, then Ψ is greater than zero and can be used to estimate relative nearness. In other words,

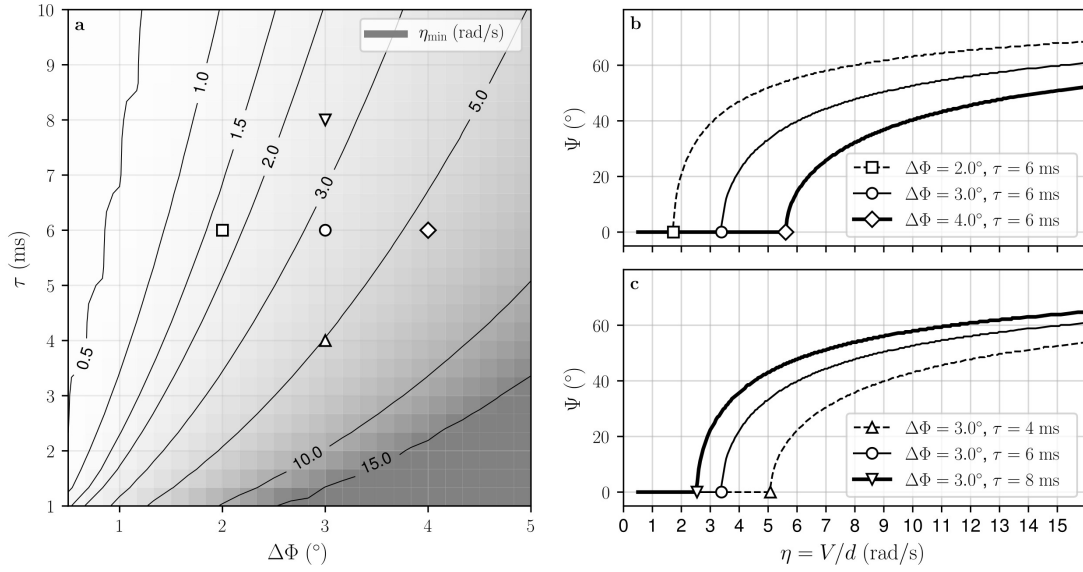


Figure 2.5 – Effect of eye resolution and EMD integration time on η_{\min} threshold and Ψ angle.

(a): The threshold η_{\min} is defined as the minimum V/d ratio above which Ψ can be used to estimate relative nearness (i.e. $\Psi > 0^\circ$). It is presented as a function of the inter-ommatidial angle $\Delta\Phi$ and the time constant τ of the EMD low pass filter blocks. η_{\min} increases for increasing $\Delta\Phi$ and it decreases for increasing τ . (b) and (c): The Ψ angle is presented as a function of relative nearness computed geometrically as $\eta = TOF_{90} = V/d$. Ψ is null for low relative nearness values ($\eta < \eta_{\min}$ in left portion of the graphs). When $\eta > \eta_{\min}$ (right portion of the graphs), Ψ is monotonically increasing with increasing relative nearness, and it can be used as an estimate of relative nearness. The shape of the curve $\eta \mapsto \Psi$ is preserved for varying values of time constant τ and inter-ommatidial angle $\Delta\Phi$. (b): For increasing value of $\Delta\Phi$, the curve is shifted to the right, i.e. to larger relative nearness. (c): For increasing value of τ , the curve is shifted to the left, i.e. to lower relative nearness.

the agent needs to fly sufficiently fast and/or close to the surface to get a relative nearness estimate from Ψ .

For Ψ to be measurable in a practical implementation, the maxima of the EMD response have to be sufficiently separated (Fig. 2.4i). The higher the relative nearness, the easier it is to detect the maxima, as shown by the relative difference between the maximum EMD response R_{\max} and the EMD response between the maxima R_{90} (Fig. 2.4j). For example, for a relative nearness of $\eta = 5 \text{ rad}\cdot\text{s}^{-1}$, our model predicts approximately 8% difference between R_{\max} and R_{90} . This value increases to approximately 22% for $\eta = 10 \text{ rad}\cdot\text{s}^{-1}$ (Fig. 2.4k).

The threshold η_{\min} depends on the time constant τ of the EMD low pass filters and on the inter-ommatidial angle $\Delta\Phi$ (Fig. 2.5). η_{\min} decreases with increasing time constant τ and

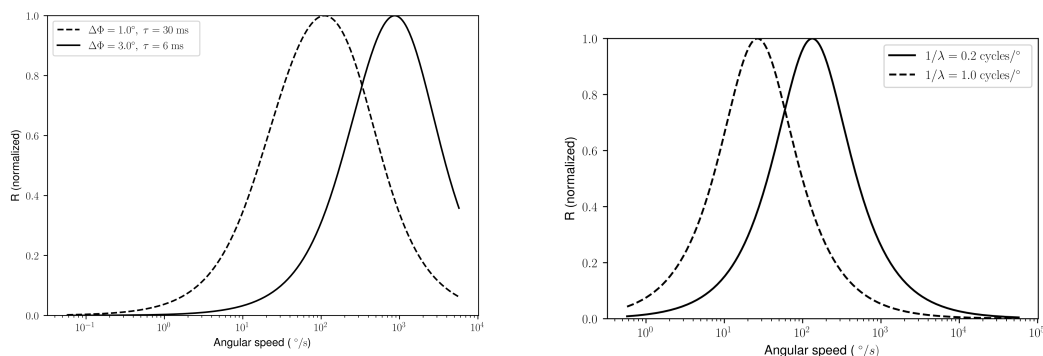


Figure 2.6 – EMD velocity response curve for a broadband signal and sinusoidal signal **(left)**: The input signal has a $1/f^2$ distribution of spatial frequencies. The time constant and the interommatidial angle of the EMD are set respectively to $\tau = 6$ ms and $\Delta\Phi = 3^\circ$ (solid curve), and $\tau = 30$ ms and $\Delta\Phi = 1^\circ$ (dashed curve). The data is computed at viewing angle $\Phi = 90^\circ$ by varying flight speed V at fixed distance $d = 0.1$ m from the pattern considered in the chapter. **(right)**: The sinusoidal signals have apparent angular frequencies of $1/\lambda = 0.2$ cycles/ $^\circ$ and $1/\lambda = 1.0$ cycles/ $^\circ$. The time constant of the EMD is set to $\tau = 6$ ms and the interommatidial angle to $\Delta\Phi = 3^\circ$. The data is computed by varying flight speed V at fixed distance $d = 0.1$ m from sinusoidal gratings with spatial period Λ chosen in order to obtain the desired apparent angular periods λ at viewing angle $\Phi = 90^\circ$.

increases with increasing inter-ommatidial angle $\Delta\Phi$. For example, an agent with an inter-ommatidial angle of $\Delta\Phi = 3.0^\circ$ and an EMD low pass filter constant $\tau = 10$ ms will have a threshold $\eta_{\min} = 2$ rad. s^{-1} (Fig. 2.5). To estimate relative nearness from Ψ (i.e. $\Psi > 0^\circ$), this agent must fly at a speed of $V > 2$ m/s when it is at a distance $d = 1$ m from the surface. Similarly, it must remain at a distance of $d < 0.5$ m when flying at a speed of $V = 1$ m/s.

2.3.4 Robustness to single-frequency pattern

The data shown here should be compared to that shown in the main text (Fig. 3, titled "Comparison of raw EMD response and Ψ angle as estimators of relative nearness"). The main text considers broadband images consisting as a sum of sinusoidal components with spatial frequencies in the range $[f_{\min}, f_{\max}]$. Here we reproduced the same study for a single-frequency image, i.e. a sinusoidal grating. The spatial period of the sinusoidal grating is $\Lambda = 5$ cm, which is in the range of spatial periods used in the main text. We can see that Ψ no longer encodes relative nearness **(h)**. Also it is notable that contrary to the broadband case, R_{\max} is mostly independent from the distance **(c)**, and Ψ is mostly independent from the flight speed **(d)**.

Chapter 2. Insect-inspired estimation of optic flow

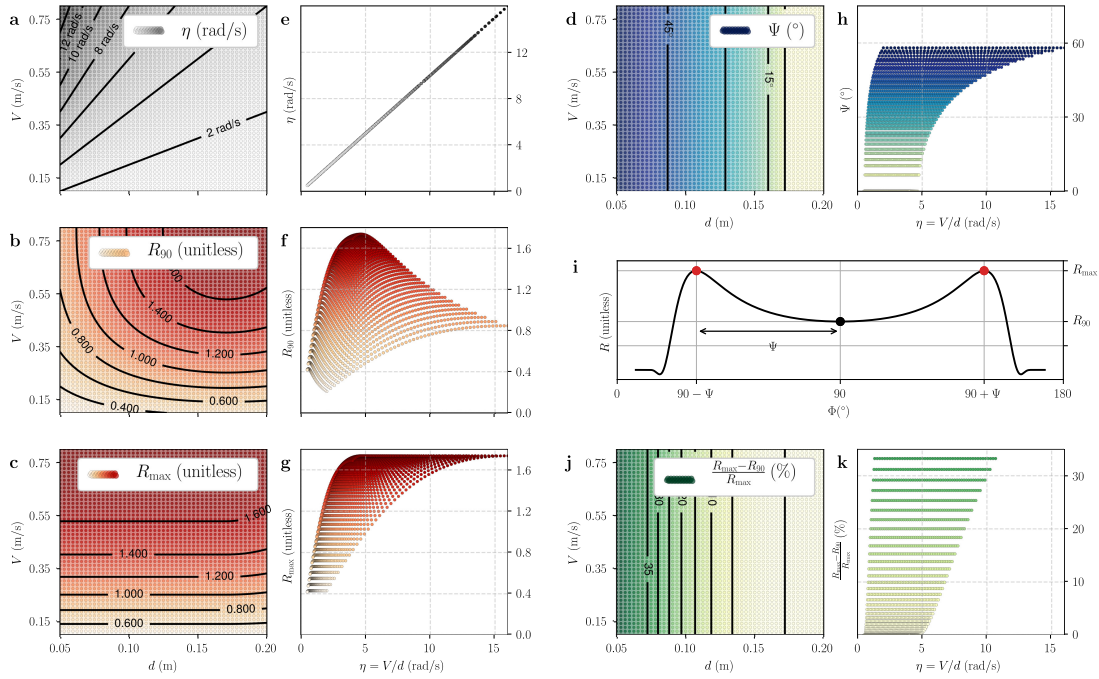


Figure 2.7 – EMD response and Ψ angle for single-frequency pattern. **(a)** and **(e)**: Relative nearness, computed geometrically as $\eta = V/d$. The unit is rad/s because this is equivalent to the angular image speed. **(b)** and **(f)**: The EMD response at 90° is defined as $R_{90} = R(\Phi=90^\circ)$. **(c)** and **(g)**: The maximum EMD response is defined as $R_{\max} = R(\Phi=\Phi_{\max})$. **(d)** and **(h)**: Deviation of the location of maximum EMD response $\Psi = |\Phi_{\max} - 90^\circ|$. Left **(a-d)**: Values given as functions of flight speed V and distance d . Right **(e-h)**: Values given as functions of the relative nearness η which is equivalent to the translational optic flow at viewing angle 90 degrees $\eta = TOF_{90} = V/d$. In all plots, the inter-ommatidial angle and time constant of the low pass filter are kept constant at $\Delta\Phi = 3^\circ$ and $\tau = 10$ ms. **(i)**: Graphical representation of R_{90} , R_{\max} and Ψ on EMD response R shown as function of viewing angle Φ . **(j-k)**: Relative difference between R_{\max} and R_{90} (given in percents), it indicates the maximum level of noise allowing the two maxima to be detected.

2.4 Simulation results

The proposed relative nearness estimator based on EMD is validated with closed-loop control of the lateral position and forward velocity of a simulated agent flying in a corridor with walls patterned by the surface shown in Fig. 2.1a. The agent can increase and decrease its forward velocity and lateral velocity. We will use the terms "forward command" and "lateral command" to refer to the velocity increments added to the forward and lateral velocity, respectively, in order to stay at equal distance to the two walls and to stabilize forward velocity at a constant value.

It is important to note that, in this section, we do not rely on the theoretical predictions of the EMD response presented in the previous section. We implemented the EMD model shown in Fig. 2.1b and computed its response to simulated images. The theoretical predictions only considered the steady-state EMD response to a signal with known power spectrum, while the results of this section use the actual response of the EMD model to computer-generated images.

2.4.1 Control strategy for lateral position and forward velocity

The control strategy is similar to those presented in previous studies [15, 98, 92, 61]. As the agent moves forward, translational optic flow (*TOF*) is computed on its left and right sides. The difference between translational optic flow on each side is used to control the lateral position of the agent. For example, a higher translational optic flow on the right side of the agent will result in a leftward command. For speed control, the average translational optic flow on the left and right sides is compared to a reference value TOF_{ref} . The agent will accelerate when the measured average translational optic flow is lower than the reference value, and decelerate otherwise. This control strategy can be summarised as

$$\begin{aligned} u_{\text{lat}} &= K_{\text{lat}} (TOF_{\text{left}} - TOF_{\text{right}}) \\ u_{\text{for}} &= K_{\text{for}} \left(TOF_{\text{ref}} - \frac{(TOF_{\text{left}} + TOF_{\text{right}})}{2} \right) \end{aligned} \quad (2.15)$$

where u_{lat} is the lateral command, u_{for} is the forward command, K_{lat} and K_{for} are proportional gains, TOF_{left} and TOF_{right} are respectively the translational optic flow measured on the left and right sides of the agent, and TOF_{ref} is a reference value. As the forward velocity is controlled using a reference translational optic flow value, the resulting forward velocity is expected to increase with increasing width of the corridor to compensate for the decreasing optic flow on the left and right sides.

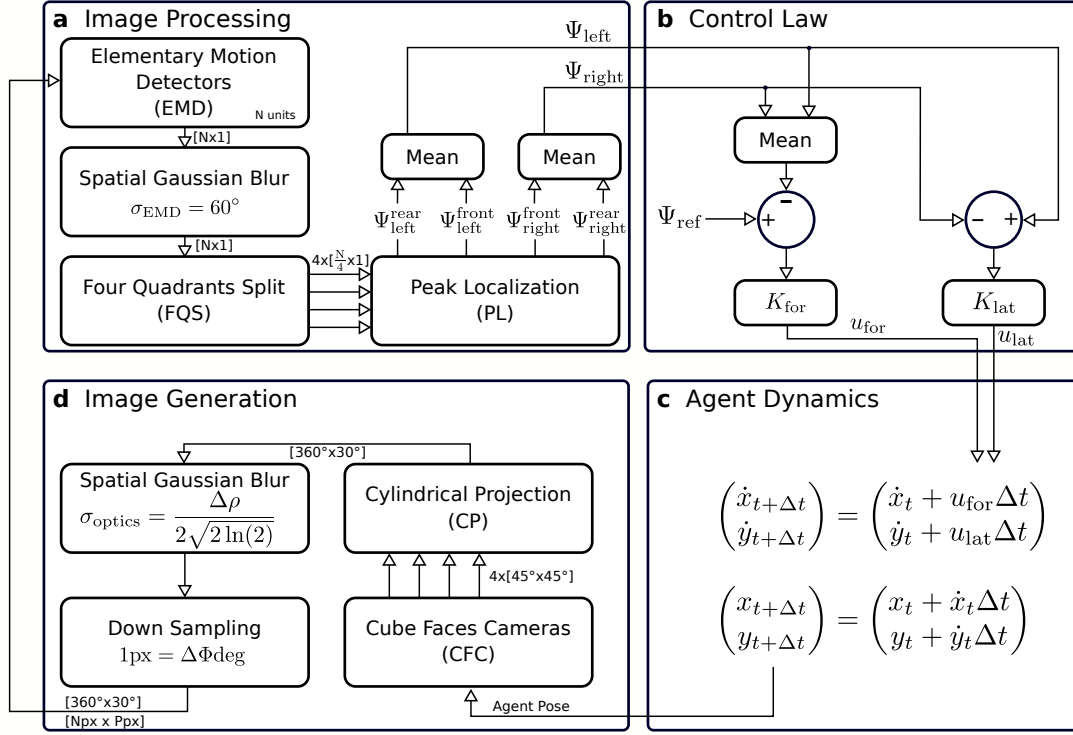


Figure 2.8 – Block diagram of the simulation.

(a): An array of N EMD units take input from consecutive pixels in the central row of a panoramic image with 360° horizontal field of view. The EMD output is spatially filtered using a gaussian kernel (with sigma σ_{EMD}) to remove spikes and ease the detection of local maxima. Four EMD output maxima are located (PL) on each quadrant (FQS), which yields Ψ values for the rear-left, front-left, front-right and rear-right quadrants. Ψ_{left} and Ψ_{right} are obtained by taking the mean of Ψ values in the left and right hemispheres. (b): Forward command u_{for} and lateral command u_{lat} are computed according to equation (2.16). The agent is pushed towards the right ($u_{lat} > 0$) when Ψ_{left} is greater than Ψ_{right} . The agent accelerates ($u_{for} > 0$) when the reference value Ψ_{ref} is greater than $(\Psi_{left} + \Psi_{right})/2$. (c): Agent dynamics are simulated as a point-mass system, where the forward velocity \dot{x} and lateral velocity \dot{y} are incremented using the forward and lateral commands u_{for} and u_{lat} . (d): Four cameras (CFC) capture images that can be mapped on the faces of a cube surrounding the agent. The cameras have a field of view of $45^\circ \times 45^\circ$, are located at the agent position and are pointed at headings 0° , 90° , 180° and 270° . The cylindrical projection block (CP) converts the four cube-face images to a single image covering a field of view of $360^\circ \times 30^\circ$ with all pixels on a row spanning a constant horizontal field of view. Spatial gaussian blur and sub-sampling are applied on the panoramic image to account for insect optics. The gaussian window has a sigma σ_{optics} defined by the acceptance angle $\Delta\rho$ of ommatidia. The image is down-sampled so that pixels point at directions separated by an angle equal to the inter-ommatidial angle $\Delta\Phi$. (a-d): In our experiments we used a simulation time-step $\Delta t = 5$ ms, cube-face images with resolution 1024×1024 pixels and inter-ommatidial angle $\Delta\Phi = 1^\circ$, leading to a panoramic image with a resolution of 360×30 pixels and $N = 360$ EMD units. The time constant of the EMD low pass filters is $\tau = 10$ ms.

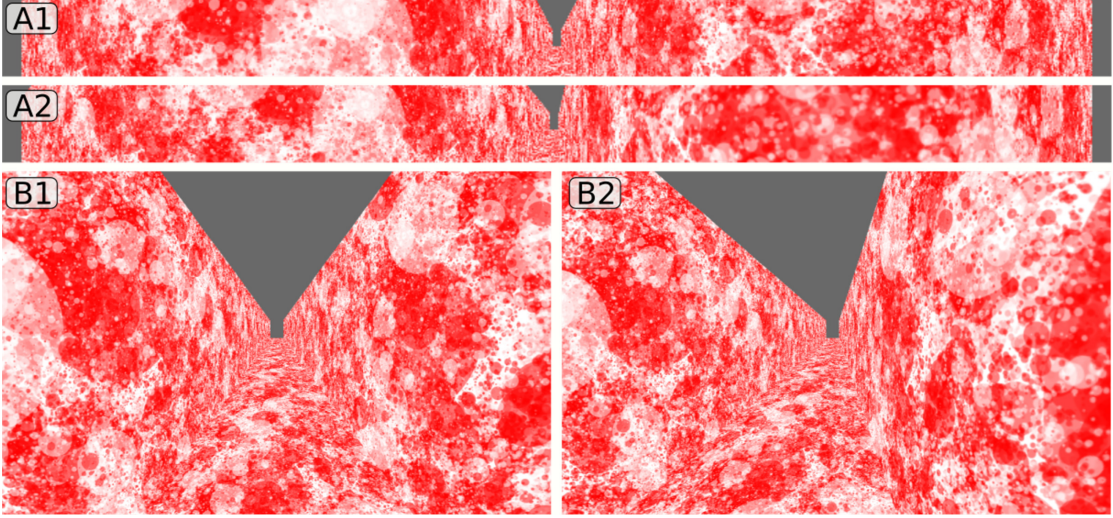


Figure 2.9 – Simulation environment.

A1 and A2: Images generated by a camera with cylindrical projection used by our simulated agent. The cylindrical projection camera has a lateral field of view of 360 degrees and vertical field of view of 30 degrees. A1 is an view from the center of the corridor. A2 is a view with the agent flying closer to the right wall. B1 and B2: Images generated using a forward looking fish eye camera used for reference only. The simulated agent does not use these images. The image B1 is taken from the same location as the image A1, similarly the image B2 is taken from the same location as the image A2.

In our experiments, the translational optic flow values TOF in equation (2.15) are replaced with the measured Ψ values:

$$\begin{aligned} u_{\text{lat}} &= K_{\text{lat}} (\Psi_{\text{left}} - \Psi_{\text{right}}) \\ u_{\text{for}} &= K_{\text{for}} \left(\Psi_{\text{ref}} - \frac{(\Psi_{\text{left}} + \Psi_{\text{right}})}{2} \right) \end{aligned} \quad (2.16)$$

where Ψ_{left} and Ψ_{right} are the deviation angles of the maximum EMD response on the left and right sides, and Ψ_{ref} is a reference value (Fig. 2.8b).

2.4.2 Simulation environment

The simulated environment consists of two vertical walls covered with a "dead leaves" pattern [137, 77] (Fig. 2.10), which contains a naturalistic distribution of spatial frequencies. The simulation can be divided in four main steps: Image processing, Control Law, Agent Dynamics and Image Generation (Fig. 2.8). At each simulation time step, a new panoramic image with 360° field of view and inter-pixel angle $\Delta\Phi$ is generated (Fig. 2.8d). The array of N EMD units takes input from consecutive pixels of the panoramic image, i.e. with constant inter-

ommatidial angle, like our eye model (Fig. 2.1a). The EMD units are updated and spatially filtered, then Ψ values are computed on left and right sides from the output of the EMD units (Fig. 2.8a). Control commands for lateral and forward acceleration are then computed from Ψ values (Fig. 2.8b). Finally, the position and velocity of the simulated agent are updated based on its current state and applied control commands (Fig. 2.8c). The four simulation steps are repeated until the agent converges to stable flight speed and lateral position.

2.4.3 Simulation results

Simulated flights were performed with different initial lateral position, initial forward speed, tunnel width and reference command Ψ_{ref} . The agent state was measured after it stabilised its velocity and lateral position (Fig. 2.11).

The agent converges towards the center of the corridor (lateral position equal zero) for each initial lateral position and forward velocity tested (Fig. 2.11a-b-c-d). The final forward speed increases with increasing tunnel width (Fig. 2.11g). This is an expected behaviour and matches the optic flow-based centering and speed control behaviour observed on flying insects. Indeed, this increase in speed allows the agent to maintain a constant optic flow for all tunnel widths (Fig. 2.11o). The Ψ angle converges to Ψ_{ref} (Fig. 2.11i-j-k-l), although it does so less reliably for lower Ψ_{ref} values (Fig. 2.11l-left). Similarly, there is higher standard deviation of the lateral position for lower Ψ_{ref} (Fig. 2.11d-left). The relationship between Ψ_{ref} and relative nearness (Fig. 2.11p) is similar to the one predicted by our analytical model (Fig. 2.4h). This confirms that Ψ is a correct estimate – though non-linear – of relative nearness.

An example of the EMD response during a simulation is shown in Fig. 2.10. At the beginning of this experiment (Fig. 2.10a), the agent is closer to the right wall and is flying at low speed. The Ψ angles are, on average, lower than the command $\Psi_{\text{ref}} = 60^\circ$, which will push the agent to accelerate. Also Ψ angles are larger on the right side than on the left side, which will push the agent towards the left, i.e. closer to the center of the corridor. This is expected because the distance to the right wall is smaller than the distance to the left wall, so the relative nearness is higher on the right wall. Note that the raw EMD response has the inverse property: the EMD response at $\Phi = +90^\circ$ (right) is smaller than the EMD response at $\Phi = -90^\circ$ (left). Thus, if our controller had used EMD response R_{90} to compute the lateral command instead of Ψ , the agent would have been pushed even more to the right and would have eventually collided into the surface. At the end of this experiment (Fig. 2.10b), the agent is flying closer to the center of the corridor with an increased speed. The Ψ angles are all close to the command $\Psi_{\text{ref}} = 60^\circ$. The agent has converged to stable lateral position and speed.

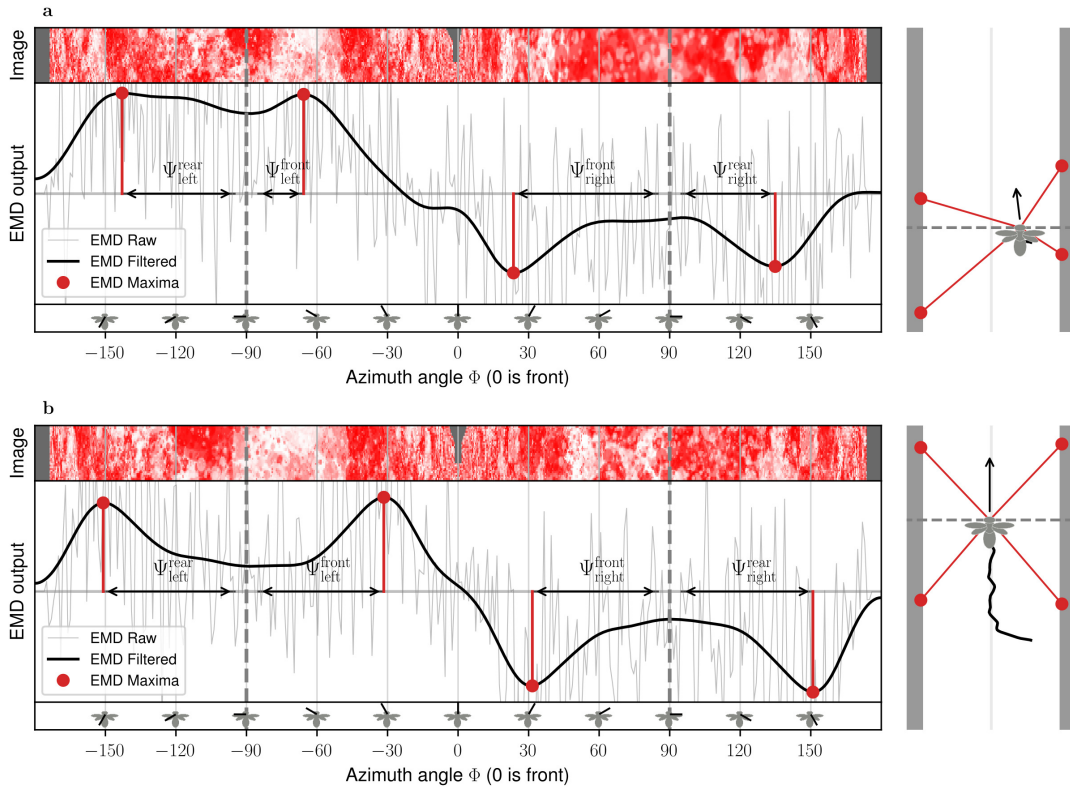


Figure 2.10 – EMD Response to Simulated Images.

The response of the array of EMDs to computer-generated images are shown at the beginning of an experiment (a), and at the end of the experiment after the agent's speed and position converged (b). The input image is shown at the top of the graph. The image is panoramic and extends from azimuth angles -180° to 180° , with the center of the image (azimuth 0°) being the front of the agent. The raw response of the array of EMDs to simulated images is represented in light grey as a function of the azimuth angle. The thick grey curve represents the signal after spatial filtering with a gaussian kernel with $\sigma_{\text{EMD}} = 60^\circ$. The maximum EMD response in each of the four quadrants are shown as red dots. The deviation of the maximum EMD output in each quadrant ($\Psi_{\text{left}}^{\text{rear}}$, $\Psi_{\text{left}}^{\text{front}}$, $\Psi_{\text{right}}^{\text{front}}$ and $\Psi_{\text{right}}^{\text{rear}}$) is measured between the maximum EMD output (red dots) and the side marks at azimuth -90° and $+90^\circ$ (grey vertical dashed lines). The drawings on the right side show the corridor seen from the top, with the position of the agent, its current speed vector and the position of the EMD maxima.

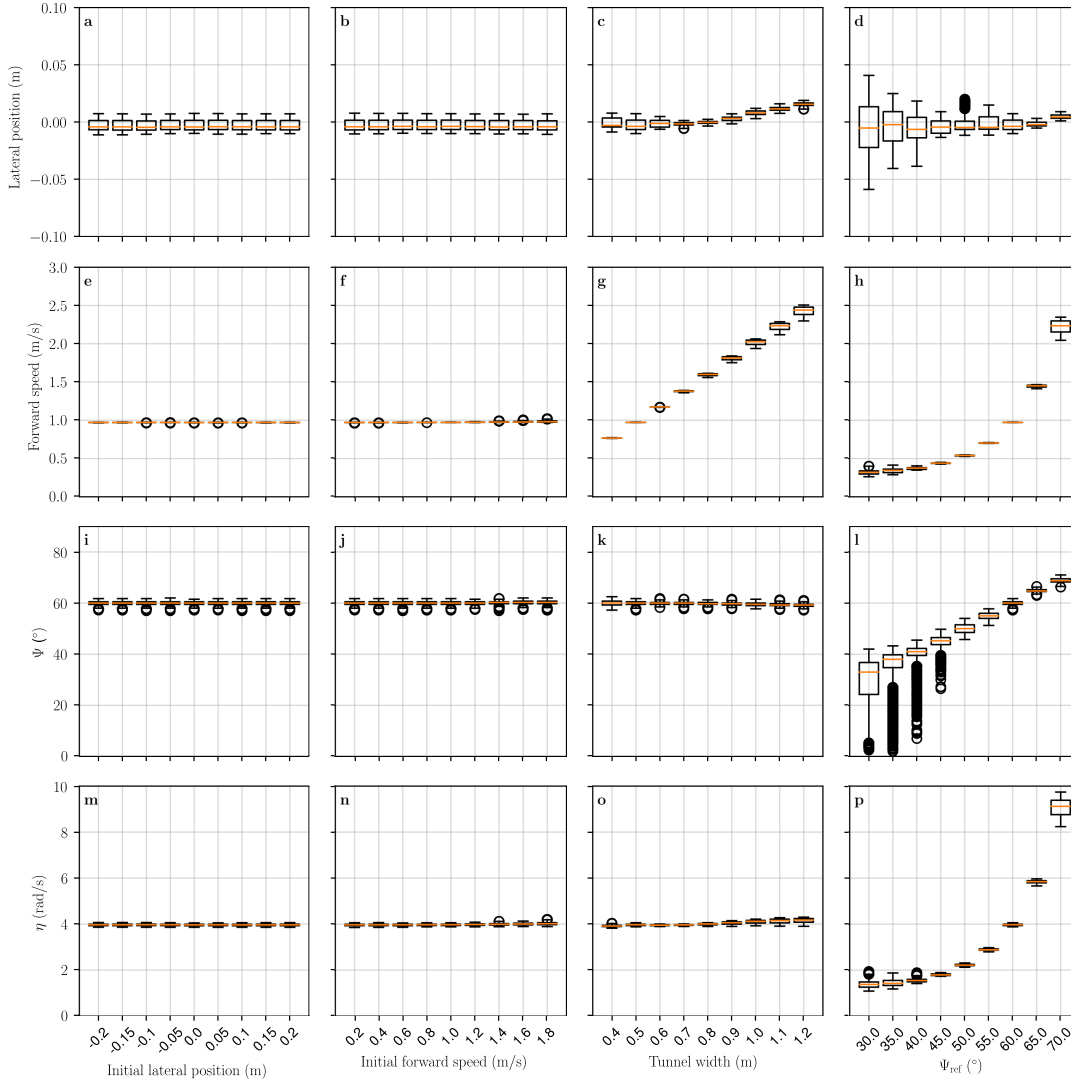


Figure 2.11 – State of simulated agent after convergence.

The boxplots are generated from the last 5 seconds of 10 second long flights. The first row (a-b-c-d) shows the final lateral position of the agent, 0 being the center of the corridor. The second row (e-f-g-h) shows its final forward velocity. The third row (i-j-k-l) is the measured averaged deviation of maximum EMD response $(\Psi_{left} + \Psi_{right})/2$. The fourth row (m-n-o-p) is the relative nearness computed as $\eta = V/D$, where V is the forward speed and D is the distance to the closest wall. In the first column (a-e-i-m), the initial lateral position of the agent is varied from 20 cm on the left to 20 cm on the right. The second column (b-f-j-n) shows results for varying initial forward speed. In the third column (c-g-k-o) the tunnel width is varied. In the fourth column (d-h-l-p) the commanded reference deviation of maximum EMD response Ψ_{ref} is varied. When not explicitly listed on the horizontal axis, the default initial lateral position is 0.1 m, the initial forward velocity 1 m/s, the tunnel width 0.5 m and the reference Ψ value is 60 degrees.

2.4.4 Robustness to non-planar scenes

Our theoretical model is derived for the case of straight flights parallel to a flat surface, hence it cannot predict Ψ values in non-planar scenes. However the robustness of our proposed relative nearness estimation technique and associated control method can be tested in more realistic scenarios. To this effect, we run the same simulation as in section 2.4.3, with the addition of obstacles on the path of the agent. The obstacles occupy 30% of the corridor width. The initial lateral position is 0.0 m, the initial forward velocity 1.5 m/s, the corridor width is 0.75 m and the reference Ψ_{ref} value is 60 degrees (Fig. 2.12).

When obstacles are placed on both sides (Fig. 2.12-bottom left), the agent stays in the center of the corridor. Approximately 1 meter before reaching the obstacles (i.e at longitudinal position of 2 meters), the agent starts to slow down. Its speed then stabilises at a value slightly above 0.5 m/s, which is coherent with the 60% reduction in corridor width. Approximately 0.5 meter before reaching the end of the narrow section, the agent starts to increase its speed until it recovers its original speed.

As it approaches an obstacle placed on the left side of the corridor (Fig. 2.12-top right), the agent slows down and moves towards the right the corridor in order to regulate Ψ values on left and right sides to Ψ_{ref} .

When six obstacles are placed alternatively on the left and right sides of the corridor (Fig. 2.12-bottom right), the agent slows down when approaching the obstacles, and it steers left and right in order to stay away from the obstacles and avoid collisions.

2.4.5 Robustness to single-frequency patterns

In order to complement the theoretical predictions made in 2.3.4, we evaluate the robustness of our control method to varying pattern properties. We reproduce the experiment presented in Fig 2.9 and Fig 2.11 that consider broadband images consisting as a sum of sinusoidal components with spatial frequencies in the range $[f_{\text{min}}, f_{\text{max}}]$. Here, we replaced the broadband images by a single-frequency image, i.e. a sinusoidal grating (Fig. 2.13). The initial lateral position is 0.1 m, the initial forward velocity 0.5 m/s, the corridor width is 0.4 m and the reference Ψ_{ref} value is 60 degrees.

In the single-frequency case, our proposed control method allows the agent to stabilise its lateral position in the center of the corridor (Fig. 2.13-top left). This is expected from the prediction (showed in Supplementary Figure S5) that Ψ decreases with the distance to the walls.

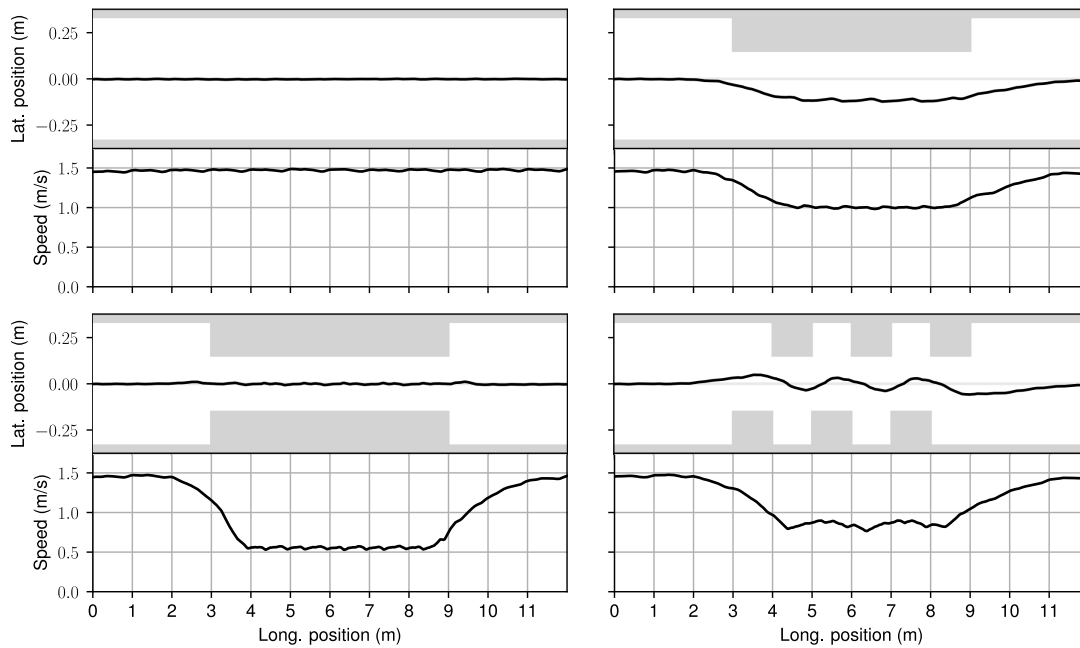


Figure 2.12 – Simulated trajectories in non-planar scenes.

(top left): No obstacle. This is the case presented in the main text. The agent flies at constant speed of 1.5 m/s in the center of the corridor. **(bottom left)**: Two obstacles are placed on both sides of the corridor. **(top right)**: One obstacle is placed on the left side of the corridor. **(bottom right)**: Six obstacles are placed alternatively on the left and right sides of the corridor in order to form a slalom.

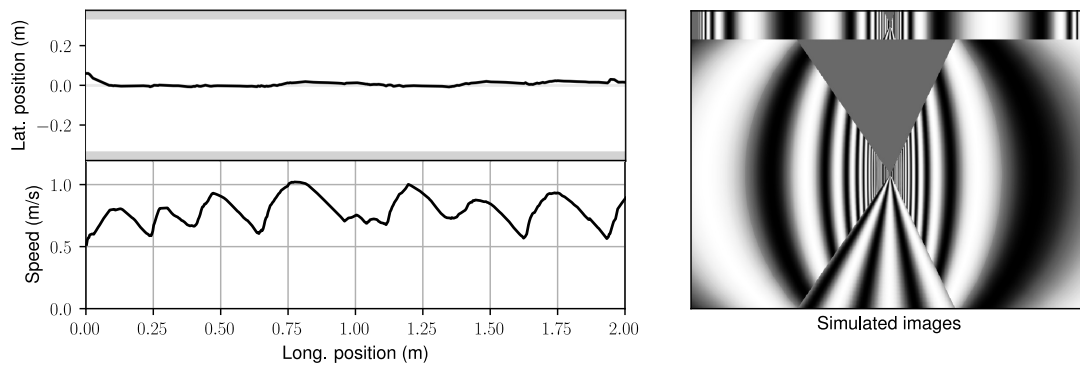


Figure 2.13 – Simulated trajectory with single-frequency pattern.

(right): The walls of the corridor are covered with a sinusoidal grating with linear period $\Lambda = 33$ cm. **(top left)**: Lateral position of the simulated agent along the corridor. **(bottom left)**: Flight speed of the simulated agent along the corridor.

However, the agent is not able to stabilise its flight speed, it varies erratically between 0.5 m/s and 1.0 m/s (Fig. 2.13-bottom left). This is expected from the prediction (showed in Supplementary Figure S5) that Ψ is independent from the flight speed.

It should be noted that a natural setting would realistically not include single-frequency patterns. Furthermore insects are likely to control flight speed using additional regions of the visual field (like the ventral and dorsal regions) that may contain patterns with richer frequency content.

2.5 Robotic implementation

The proposed algorithm for estimation of relative nearness is validated in real world conditions on a flying robot (Fig. 2.14a).

2.5.1 Hardware setup

The algorithm is implemented on a smart camera, the PX4Flow [59], which contains on a single board a global shutter CMOS sensor and an STM32F4 microcontroller to perform signal processing. The camera was adapted with a fish-eye lens in order to provide a field of view of 180° with 620 pixels laterally. The vertical field of view was limited to 30° , i.e. the camera samples elongated panoramic images. The camera is calibrated and ommatidias with acceptance angle $\rho = 1^\circ$ and an inter-ommatidial angle $\Delta\Phi = 1^\circ$ are emulated by applying a gaussian blur and by sub-sampling the image, as described for the simulated experiments (Fig. 2.8).

The microcontroller of the smart camera computes EMDs outputs at 200 frames per second, with time constant $\tau = 10$ ms locates the maximum response, and transmits this information to the main autopilot over a serial connection. With this architecture, only high level information is transferred from the camera module to the autopilot, which is offloaded from image processing operations.

The flying robot is a fixed wing equipped with a MAV'RIC autopilot (see Appendix B) for attitude stabilization, and with a GPS and a pitot tube in order to obtain maintain constant flight speed. The camera is mounted below the fixed wing drone, pointed down, so that the rightmost part of the image points forward and the leftmost part of the image point backward (Fig. 2.14b). The robot is flown at speed $V \approx 15$ m/s and altitude above ground d varying between 0.5 meter and 5 meters. The EMD output, measured Ψ angle, and the ratio between flight speed and altitude above ground are logged on an SD card.

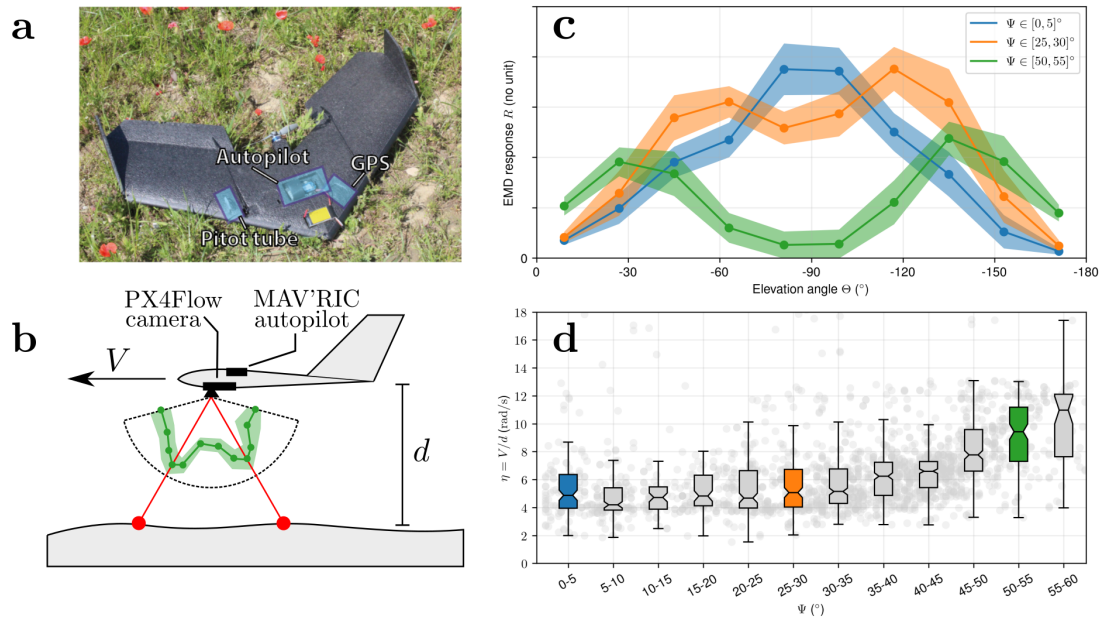


Figure 2.14 – EMD response and relative nearness estimation onboard fixed wing drone. (a): Fixed wing robot used for the experiment. The robot has a wingspan of 0.8m and a weight of 0.6kg. It is equipped with a MAV'RIC autopilot for state estimation, flight control and data logging. (b): A customized PX4Flow camera module is mounted below the fixed wing drone and equipped with a fisheye camera providing a wide field of view of 180° oriented front to back on the aircraft. (c): EMD response as a function of the location in the visual field. Given the orientation of the camera, the location is expressed using the elevation angle Θ which is equal to 0° in the forward direction, -90° in the viewing direction pointed vertically downward, and -180° in the backward direction. Colored solid lines represent the mean EMD responses in 10 segments of the visual field. The color shaded areas represent the standard deviation. Similarly to the theoretical and simulated results, the EMD response presents two maxima for higher values of relative nearness. (d): Relative nearness as function of the measured Ψ angle. The relative nearness is computed from the logged flight speed and altitude above ground. This graph shows similar trend to the theoretical study (Fig. 2.4h), and to the simulation (Fig. 2.11p).

2.5.2 Flight results

The response of the EMD array recorded during flight displays the same trend as the theoretical response (Fig. 2.3a) and the simulated response (Fig. 2.9b), i.e. it has either a single maximum located at 90° from the direction of motion when flying, or two maxima depending on the relative nearness $\eta = V/d$ (Fig. 2.14c).

The angular separation between the two maxima varies with the relative nearness in a similar fashion to the theoretical results (Fig. 2.4h) and the simulated results (Fig. 2.8p). The angle Ψ is close to 0 for values of relative nearness below a threshold η_{\min} which is around 5 rad/s with the selected EMD parameters (Fig. 2.14d, left), and it increases with increasing values of the relative nearness. Due to the noisy signal, denoted by the large standard deviation of the relative nearness, the relative nearness estimated by Ψ does not evolve for low Ψ values, and becomes useable for Ψ values above approximately 45° (Fig. 2.14d, right).

2.6 Discussion

The EMD is a biological model for motion estimation that has received strong experimental support as the foundation of motion detection in insects. Due to its relative simplicity, the EMD model also has good potential as a computationally fast motion estimator for engineering applications. Indeed, an EMD requires two multiplications for each pixel, one subtraction and two time delays while the Lucas Kanade algorithm [88] requires 11 multiplications and 6 subtractions. However, the EMD model output does not provide a perfect estimation of relative nearness as it cannot be unambiguously expressed in angular speed. The response of EMDs for varying angular velocity indeed has a bell shape with a maximum at an angular image velocity that is function of the EMD parameters as well as the spatial frequency of the input signal. This is problematic for biologists because insects appear to rely on relative nearness for flight control [124, 40] independently of image properties. The ambiguous nature of the EMD output is also problematic for engineers who require measures of angular speed (expressed in pixels or radians per second) for tasks such as ego-motion estimation or mapping. Also, for larger angular velocities, the EMD response decreases in a way that cannot be discriminated from a decrease in angular velocity. For example, as an agent approaches a surface – and thus as the angular image velocity increases – the response of an EMD may start decreasing. This leads to a crash into the surface when the EMD response is used to compute a repulsive force [84, 14]. This case is shown in Fig. 2.10a where the absolute value of R_{90} is larger on the left side of the agent than on the right side even though the agent is offset to the right of the corridor – i.e. the angular image speed is smaller on the left side of the agent. Our simulated agent would have crashed into the wall on its right if the EMD response was used instead of Ψ

to compute its lateral command.

In other words, there is an apparent incompatibility between the main neurophysiological model for motion estimation (EMD) and the main behavioural model for insect flight control (optic flow). Several studies have proposed modifications to the EMD in order to correct its output (for example [135, 57, 23, 78]). Although they demonstrate improved robustness to varying contrast and spatial frequency, these models often require additional computational blocks. Most importantly, these models are less well-supported by electrophysiological recordings from the insect visual system. Here, we have shown that it is indeed possible to use a simple Hassenstein Reichardt EMD output for estimation of relative nearness with limited additional computational blocks – namely a spatial blurring and maximum location. These blocks integrate EMD responses across the visual field without modifying the structure of the correlator [56]. Because our method relies on spatial integration across wide field of view, it is especially suited to estimation of relative nearness to a large obstacles around the agent, or to the ground below the agent.

We introduced the angle Ψ , which is the angle between the viewing direction pointing directly at the patterned surface and the viewing directions with maximal EMD response and showed that this angle is closely related to relative nearness (Fig. 2.4) and is therefore suitable for controlling flight (Fig. 2.10). Our model predicts Ψ in the limited case of straight flights parallel to a planar surface. However, we demonstrated successful flight control based on Ψ in a simulation environment that does not constrain the agent to fly along straight paths (Fig. 2.10-right), and also for non-planar scenes (see Supplementary Figure S7). The main novelty of the angle Ψ is that it relies on the relative response of several EMD detectors instead of relying on the absolute value of their output. In other words, we suggest that relative nearness is spatially encoded by the relative response of EMDs rather than by the magnitude of their responses, something that has strong biological plausibility. Indeed, computing Ψ consists mostly in detecting the maximum response in an array, which is easily implemented in neural systems using a Winner-Take-All network [111] or using differentiation and zero-crossing [97].

2.6.1 Biological relevance of η_{\min}

While there has been much behavioural evidence that honeybees use relative nearness to control their flight [4, 124], two recent studies have shown that flight control in bumblebees does indeed exhibit some dependency on the spatio-temporal properties of sinusoidal gratings [37, 28]. These apparently conflicting results can nonetheless be explained by the method we propose here because the maximum output of an array of EMDs would exhibit spatio-temporal dependencies when presented with patterns containing single frequencies (see Supplementary

Figures S5 and S8) but not when presented with more complex patterns containing multiple frequencies, such as checkerboards that contain a series of discrete frequencies that are harmonics of a fundamental spatial frequency related to the size of the checkerboard squares. To avoid the ambiguities created by sinusoidal and checkerboard patterns and to make our study more relevant to the natural behaviour of insects, we considered the output of the EMD model in response to the dead-leaves pattern [137, 77], which has a spectral content that matches that of natural scenes [131, 8] with a distribution of frequency of $1/f^2$. The method we propose in this chapter for extracting relative nearness from EMD output is a consequence of EMD dependency on spatial frequency, coupled with the geometry of the environment. The response of EMD is tuned to a specific ratio between image speed and angular period [135]. When insects fly above the ground or beside large flat objects, visual features of the environment are seen from a greater distance in the forward and rearward regions of their field of view. Hence, these features would subtend a smaller angle in the field of view, that is, they would appear to have a smaller angular period, i.e. a higher spatial frequency. As a consequence, the ratio of image speed to angular period at which the EMD output is maximal is achieved only at specific viewing angles, which then provides an estimate for relative nearness. Our scheme of using the angle Ψ to estimate relative nearness thus explains both the results that find spatio-temporal dependency of flight control behaviour and those that find optic flow dependency. Our scheme also highlights the importance of the structure of the pattern being used on the results of behavioural experiments on flight control.

Locating the maximum EMD response provides an estimate of relative nearness only above a threshold value η_{\min} . This means that Ψ is a valuable measure only if the agent is flying fast enough and/or close enough to the surface. Our model predicts the value of the relative nearness threshold η_{\min} from the inter-ommatidial angle $\Delta\Phi$ and the time constant τ of the EMD low pass filters (Fig. 2.5). We can investigate whether Ψ is a candidate for relative nearness information in an insect species from the speed over distance ratio V/d at which it flies and testing whether it is higher than the value of η_{\min} that is predicted from its interommatidial angle and time constant. For an insect such as a bee, with an inter-ommatidial angle $\Delta\Phi = 3.0^\circ$ measured [120] at an azimuth angle $\Phi = 90^\circ$ and an estimated time constant [55] $\tau = 10$ ms, the predicted threshold is $\eta_{\min} = 2.0 \text{ rad}\cdot\text{s}^{-1}$ (Fig. 2.5a). This threshold is indeed lower than the flight speed to distance ratio at which bees flew in previous experiments: lateral relative nearness was recorded between 3.0 rad/s and 3.8 rad/s in *Bombus terrestris* [7, 81], and it was recorded between 3.75 rad/s and 4.96 rad/s in *Apis mellifera* [4, 117]. This supports the hypothesis that these species may be using the visual angle at which maximal EMD output occurs to estimate relative nearness in order to control their flight speed. The same test can be replicated for other species using experimental measurements of $\Delta\Phi$, τ , and V/d .

Several studies have shown that ventral relative nearness may also be used by insects for flight control [7, 81, 82, 100], in addition to lateral relative nearness. Bumblebees rely primarily on lateral relative nearness cues for speed control when navigating narrow corridors, but ventral relative nearness cues are preferred over lateral relative nearness cues in wider corridors [81]. However, the lateral relative nearness in narrower tunnels (approx. 3.5 rad/s) is much smaller than the ventral relative nearness in wider tunnels (approx. 5.7 rad/s). Can this be explained by our model? Insect eyes tend to have reduced resolution in the ventral region [68], thus, Ψ values are expected to be lower in the ventral region than in the lateral region (Fig. 2.5b). For the narrow corridor case ($\Delta\Phi_{\text{lateral}} = 3.0^\circ$, $\tau = 10$ ms, $\eta_{\text{lateral}} = 3.5$ rad/s) our model predicts $\Psi_{\text{lateral}} = 40^\circ$ (Fig. 2.5b). Within our control strategy (Fig. 2.8b), this corresponds to the bee maintaining Ψ equal to the reference value $\Psi_{\text{ref}} = 40^\circ$. Assuming that τ is uniform across the eye, and that bees use the same reference Ψ_{ref} to control flight speed when using lateral motion cues and when using ventral motion cues, we can predict the ventral inter-ommatidial angle that matches the higher relative nearness in the ventral region. For the wide corridor case where $\eta_{\text{ventral}} = 5.7$ rad/s, $\Psi_{\text{ventral}} = 40^\circ$ is obtained with inter-ommatidial angle $\Delta\Phi_{\text{ventral}} = 4.0^\circ$ (Fig. 2.5b), which is indeed larger than $\Delta\Phi_{\text{lateral}}$. In other words, with equal Ψ_{ref} and τ in ventral and lateral regions, but with a larger ventral inter-ommatidial angle $\Delta\Phi_{\text{ventral}} = 4.0^\circ$ than lateral inter-ommatidial angle $\Delta\Phi_{\text{lateral}} = 3.0^\circ$, our model correctly predicts the lateral and ventral relative nearness measured in bumblebees [81].

2.6.2 Dynamic adaptation of the visual system

The relative nearness estimate provided by Ψ is most precise for η values slightly superior to the threshold η_{min} . Indeed the slope of the function $\eta \mapsto \Psi$ is maximum for η values just above the η_{min} threshold, i.e. a small variation in relative nearness would lead to a large variation of Ψ (Fig. 2.4h and Fig. 2.5b-c). Below values of η_{min} , however, Ψ provides no information as the slope of the function $\eta \mapsto \Psi$ is null for $\eta < \eta_{\text{min}}$ (Fig. 2.4h and Fig. 2.5b-c). As a consequence, a flying agent should control its flight in order to maintain V/d values close to the threshold η_{min} , but not below that threshold.

Conversely, adapting η_{min} to a value just below the currently experienced relative nearness value maximizes the precision of the Ψ estimate. Our results show that the value of the threshold can be adapted by varying the inter-ommatidial angle and the EMD time constant (Fig. 2.5a). A reduced inter-ommatidial angle leads to a reduced threshold (Fig. 2.5b) which would enable relative nearness estimates at low flight speed and/or faster reactions to obstacles that are approaching in the direction of flight. Inter-ommatidial angles are fixed by the anatomy of the compound eye and thus cannot be modified during flight. Nonetheless, the

distribution of inter-ommatidial angles across the eye in different insect species may reflect adaptations that better enable Ψ estimates in relevant parts of the visual field. Another way in which the EMD output can be adapted is by modifying the time constant τ , which can be dynamically varied during flight [86]. An increased time constant would lead to a decreased threshold which is desirable at low flight speed, while a decreased time constant would lead to an increased threshold which is desirable at high flight speed (Fig. 2.5c). We suggest that a flying agent using Ψ for flight control can improve the precision of its relative nearness estimate by increasing the value of the time constant τ at low speed, and decreasing its value during fast forward flight. Biological evidence for dynamic changes in τ comes from [104], who showed that the decreased flight speed in bumblebees that is observed in response to decreased light intensity is accompanied by an increased time constant in the photoreceptors.

2.6.3 Coping with EMD measurement noise

The response of the EMD array to moving images contains spikes resulting from transient responses (Fig. 2.10 light grey). Transient EMD responses are present in our simulation but not in our model that considers only the steady-state EMD response [39]. Nonetheless, transient EMD response spikes represent measurement noise which have to be dealt with. For example, spatial differentiation and zero-crossing, which is a potential neuronal implementation for maximum detection [97], would be strongly affected by such spikes. In simulation experiments, Ψ angles in the front and rear of the visual field were averaged (Fig. 2.8a), which lowers measurement noise. In addition, spatial integration of the EMD response was performed front-to-back with a spatial gaussian filter in order to remove spikes and facilitate the detection of maxima (Fig. 2.8a). However, the gaussian filter also flattens peaks around EMD maxima, which makes peaks difficult to disambiguate when maxima are close to each other (Fig. 2.10a-left), and may result in the detection of a single maxima ($\Psi \approx 0^\circ$). This is a potential explanation for the low Ψ outlier values that appear at low Ψ_{ref} angles, i.e. when EMD maxima are close to each other (Fig. 2.11-left). Whether the EMD maxima can be detected and located in the presence of noise is a fundamental requirement for the applicability of our method in a real world scenario. Our model predicts the difference between the peak EMD response and the EMD response at $\Phi = 90^\circ$, i.e. the EMD response in the "well" between the two maxima (Fig. 2.4i-k). We showed that for low relative nearness values above the threshold η_{min} , the EMD maxima are not only close form each other (small Ψ on the left of Fig. 2.4h), but also separated by a well of similar amplitude (R_{90} close to R_{max} on the left of Fig. 2.4k). Figure 2.4k shows the maximum level of measurement noise that allows the EMD maxima to be distinguished from the EMD response at 90 degrees. To obtain a reliable estimate of relative nearness using the EMD output, it is not only necessary to keep the V/d value above

the η_{\min} threshold, but it is also necessary to keep a margin above η_{\min} in order to have clearly separated peaks in the EMD response (Fig. 2.10b). Biological data discussed previously [81, 7, 117, 4] suggest that bees fly with $\Psi_{\text{ref}} = 40^\circ$, which means that EMD maxima would be separated by a comfortable angle of 80° .

2.6.4 Bi-dimensional field of view

How does our method for estimation of relative nearness generalize to a two dimensional field of view? With a 2D spherical field of view, a viewing direction is defined by its elevation angle Θ in addition to the azimuth angle Φ . The present study assumes an elevation angle equal to zero. In equation (2.11), the apparent temporal frequency $\hat{\omega}$ is not affected by varying elevation angle, the same way it is not affected by varying azimuth angle. The geometry of the environment is axially symmetric about the viewing direction with azimuth $\Phi = 90^\circ$ and elevation $\Theta = 0^\circ$ (Fig. 2.1a) and so is the apparent angular period $\hat{\lambda}$ and apparent signal amplitude $\hat{\Delta I}$. As a consequence, the EMD output R is constant over circles centered on the viewing direction $\begin{bmatrix} \Phi \\ \Theta \end{bmatrix} = \begin{bmatrix} 90 \\ 0 \end{bmatrix}$. In other words, with a 2D EMD array, there are not only two maxima in the EMD response. Instead, there is a circle where the EMD response is maximal. This circle is centered on viewing direction $\begin{bmatrix} \Phi \\ \Theta \end{bmatrix} = \begin{bmatrix} 90 \\ 0 \end{bmatrix}$, and the radius Ψ of the circle can be used as an estimate of relative nearness. Note that this circle contains the two viewing directions with maximum EMD response presented in this study ($\begin{bmatrix} \Phi \\ \Theta \end{bmatrix} = \begin{bmatrix} 90-\Psi \\ 0 \end{bmatrix}$ and $\begin{bmatrix} \Phi \\ \Theta \end{bmatrix} = \begin{bmatrix} 90+\Psi \\ 0 \end{bmatrix}$) and is thus an extension of the 1D case. The immediate benefit of using two dimensional field of view is the reduction of measurement noise by spatial integration. Instead of performing spatial integration front-to-back (Fig. 2.8a), it could be performed using the second dimension of the image — along circular paths — in order to measure the average radius Ψ of the circle where EMD response is maximum. We expect that with this method, maxima would be easier to distinguish even at low Ψ angles, while still filtering transient EMD response spikes. Whether this two dimensional spatial integration is used by insects could be tested by checking whether the output of motion detecting neurons are spatially pooled across regions with circular shape.

2.7 Conclusion

In this chapter, we modeled the response of an array of EMDs in the case of an agent flying along a flat patterned surface and showed that the raw value of the EMD response is poorly correlated to relative nearness. We showed that the location of the maximum response in the EMD array provides appropriate estimation of relative nearness when the agent is flying sufficiently fast and/or close to the surface. We introduced the notion of relative nearness threshold to provide bounds on speed and distance, and showed that they are consistent

with data from flight control experiments on *Bombus terrestris* and *Apis mellifera*. Finally, we proposed a flight control strategy that uses the location of maximum EMD response as control input instead of optic flow and tested it in a 3D simulation where we successfully controlled the forward velocity and lateral position of a simulated agent flying in a corridor. Similar to what is observed in insects, and as expected with optic flow based control, the agent's forward velocity is dependent on the corridor width: the broader the corridor, the faster the agent advances.

The method of extracting relative nearness from EMD output that is described here relies on a standard form [9] and requires few additional computational steps: namely, spatial filtering and detection of maximum EMD response, with both being easily modeled as neuronal networks. Further studies are needed in order to investigate if this scheme is indeed used in biological systems and to identify its neural underpinnings. Nevertheless, our method provides an algorithm for estimation of relative nearness that has low computational cost and that could be readily used in robotics applications.

3 The role of optic flow pooling in insect visual flight control

FLIGHT through cluttered environments, such as forests, poses great challenges for animals and machines alike because even small changes in flight path may lead to collisions with nearby obstacles. When flying along narrow corridors, insects use the magnitude of visual motion experienced in each eye to control their position, height, and speed but it is unclear how this strategy would work when the environment contains nearby obstacles against a distant background. To minimise the risk of collisions, we would expect animals to rely on the visual motion generated by only the nearby obstacles but is this the case? To answer this, we combine behavioural experiments with numerical simulations and provide the first evidence that bumblebees extract the maximum rate of image motion in the frontal visual field to steer away from obstacles. Our findings also suggest that bumblebees use different optic flow calculations to control lateral position, speed, and height.

Chapter 3. The role of optic flow pooling in insect visual flight control

This chapter is adapted from the journal publication:

- [74] Lecoer, J., Dacke, M., Floreano, D. & Baird, E. The role of optic flow pooling in insect flight control in cluttered environments. *Scientific Reports* **in review** (2018).

3.1 Introduction

By rapidly processing visual information into motor commands, insects are able to navigate safely in cluttered environments with a level of miniaturisation and refinement that is unmatched by man-made systems. Collision avoidance in clutter is poorly understood in biological systems and is an active field of research for the development of miniature aerial robots [46] where standard computer vision guidance algorithms require high camera resolution and computing power, which heavily restrict flight time. Understanding and taking inspiration from the simpler – yet effective – techniques employed by insects may help engineers in their quest for miniaturisation and autonomous operation of miniature aerial robots.

Flying insects use the pattern of wide-field image motion on their retina – called optic flow to control their flight. Optic flow is measured across the panoramic field of view of their compound eyes [41, 18] from the output of arrays of visual motion detection units that extract a motion estimate from neighbouring ommatidia [56] over a small portion of the visual field. The outputs of these units are then pooled across larger parts of the visual field by integrating neurons [49, 61, 63, 14], which results in a wide-field representation of optic flow that can then be used to guide various aspects of flight behaviour such as lateral position, flight speed, and vertical position. What remains unclear is exactly how the pooling calculation that is being used to control flight is being performed and whether this is done across the entire visual field or in only select parts of it.

When flying through experimental corridors, bees appear to control their lateral position by balancing the magnitude of translational optic flow experienced in the lateral visual field of each eye [122]. Because translational optic flow varies with the inverse of distance [53, 64], this strategy minimises the risk of collision by ensuring that they maintain an equal distance to each wall. Similarly, flying insects also control their flight speed using optic flow – the magnitude of translational optic flow in the lateral, ventral or dorsal visual field [100, 82, 103] is maintained at a set-point such that, in experimental corridors, flight speed decreases with the distance between the walls [31, 123, 4, 7]. It appears that flight speed is regulated by optic flow in the lateral, ventral and dorsal visual fields [100, 6, 82]. Altitude control in flying insects is much more poorly understood. Bumblebees appear to use optic flow in both the ventral and lateral visual field for vertical position control [81] but evidence from honeybees suggests that they control altitude by maintaining ventral or dorsal optic flow at a set-point, as for speed control [99, 101]. However, without additional input, this control scheme would suffer from a scaling issue – the rate of ventral flow will remain constant if the animal makes equally proportional changes in both speed and altitude. Thus, exactly how insects control their vertical position and where they measure the optic flow used for this remains unclear.

Bees routinely forage in cluttered natural environments, such as around bushes or in forests, where the risk of colliding with obstacles is high. The obstacles that pose the greatest collision risk in clutter are those in front and would generate higher magnitudes of optic flow than the more distant background, but these would subtend only a minor portion of the visual field. If the value for the magnitude of optic flow that is used for flight control is averaged across the output of motion detectors across a broad visual field, then information about the proximity of obstacles in front of the agent will be lost. Thus, how and where in the visual field optic flow is calculated will severely affect the performance of any vision-based flight control and collision avoidance strategies. Previous work on insect flight control provides little insight into how insects measure optic flow for flight control because the pattern of optic flow generated in the experimental corridors that are typically used in these studies is qualitatively independent of viewing angle and the nearest obstacles (the walls) occur only in the lateral visual field.

To investigate how insects control flight in cluttered environments and to identify the optic flow pooling strategies they use to do so, we recorded the trajectories of bumblebees flying through corridors in which the density and placement of obstacles were varied. We then calculated the response of different optic flow pooling methods to our different experimental environments and compare this to the measured data. We find that lateral position and speed control are being regulated by an optic flow pooling strategy that extracts optic flow from nearby obstacles in the frontal visual field but that height control is being regulated by a value of optic flow derived by averaging optic flow across a wide lateral field of view. Overall, our findings suggest that rather than calculating optic flow only across a wide visual field, bees selectively react to nearby obstacles for centring and speed control and that optic flow for different flight control behaviours may be mediated by parallel processing streams in the insect visual system.

3.2 Materials and Methods

3.2.1 Experimental setting

Bumblebees (*Bombus terrestris*) were trained to fly along an experimental corridor between their nest and a food source. Bumblebees accessed the corridor through small holes and neither the nest nor the food source were visible from the inside. The experimental corridor was 3 m long, 0.6 m wide with 0.6 m high walls (Fig. 3.1). Each row of obstacles was located $d = 0.1$ m off the center-line of the experimental tunnel. The obstacles consisted of $0.03 \text{ m} \times 0.004 \text{ m} \times 0.6 \text{ m}$ vertical plates and were equally distributed along the tunnel. The spacing between the obstacles was varied between infinity (no obstacles), 0.15 m, 0.06 m, and 0 m (a

wall without gap).

The walls, floor, and obstacles were covered with a red and white pattern (to improve detection of the bee in the camera images) providing visual contrast in all orientations and thus enabling the insects to measure image motion around them [Lecoeur2018, 137, 27, 82]. This pattern – called dead leaves – has the same frequency content as natural images and is rotation and scale invariant. The scale invariance of the dead leaves pattern implies that nearby obstacles appear with the same frequency content as the more distant background. This setting ensures that the only way to distinguish the obstacles from the background is the difference of their relative motion. Obstacles can thus be distinguished from the background based on dynamic cues but not based on static cues.

The foraging flights were filmed with two high-speed cameras at 200 fps and the 3D trajectories reconstructed. The flight trajectories were mostly rectilinear along the x axis, with a constant speed V and altitude z , and minor oscillations along the lateral y axis. We therefore model bee trajectories as straight trajectories at a constant speed, parallel to the x axis.

3.2.2 Geometrical expression of translational optic flow

The magnitude of optic flow perceived by an agent when moving along an experimental corridor can be predicted geometrically [53, 64] according to equation [3.1].

$$\|OF(x, y, z, V, \Psi)\| = \frac{V \sin^2(\Psi)}{\Delta} \quad (3.1)$$

where x , y and z are the longitudinal, lateral, and vertical positions of the agent in the corridor, respectively. The lateral position is equal to 0 when the agent is in the center of the corridor and $y > 0$ when the agent is offset towards the left side of the corridor. The vertical position z is equal to zero when the agent touches the ground and $z > 0$ when the agent is above the ground. V is the forward velocity. Ψ is the viewing angle at which optic flow is measured. The angle Ψ is equal to zero for the viewing direction pointed straight ahead, and $\Psi > 0$ when looking on the left side of the agent. Δ is the projection onto the y axis of the distance between the agent and the closest object (see Fig. 3.1). In our experimental corridors (conditions $\mathcal{C}_{0|0}$ to $\mathcal{C}_{33|33}$), the distance Δ can be the distance from the agent to the left wall, right wall, left line of obstacles, or right line of obstacles (Fig. 3.1).

3.2.3 Optic flow pooling

We define optic flow pooling as the operation \mathcal{P} that consists in computing an optic flow value $OF^{\mathcal{P}}(x, y, z, V)$ that does not depend on the viewing direction Ψ from a panoramic optic flow field $OF(x, y, z, V, \Psi)$. We consider four pooling methods, noted \mathcal{P}_{avg} , $\mathcal{P}_{\text{avgf}}$, \mathcal{P}_{max} and $\mathcal{P}_{\text{maxf}}$, that differ in the extent of the field of view (either complete or frontal field of view) and in their pooling function (either average or maximum pooling) as described below.

The average optic flow pooling used in this chapter is noted \mathcal{P}_{avg} and computed according to equation [3.2] for viewing angles $\Psi \in [0, \pi]$ and $\Psi \in [-\pi, 0]$.

$$\begin{aligned} OF_{\text{left}}^{\mathcal{P}_{\text{avg}}}(x, y, z, V) &= \frac{2}{\pi} \int_0^{\pi} \|OF(x, y, z, V, \Psi)\|.d\Psi \\ OF_{\text{right}}^{\mathcal{P}_{\text{avg}}}(x, y, z, V) &= \frac{2}{\pi} \int_{-\pi}^0 \|OF(x, y, z, V, \Psi)\|.d\Psi \end{aligned} \quad (3.2)$$

The average optic flow pooling in the frontal visual field is noted $\mathcal{P}_{\text{avgf}}$ and computed according to equation (3.3) for viewing angles $\Psi \in [0, \pi/2]$ and $\Psi \in [-\pi/2, 0]$.

$$\begin{aligned} OF_{\text{left}}^{\mathcal{P}_{\text{avgf}}}(x, y, z, V) &= \frac{4}{\pi} \int_0^{\pi/2} \|OF(x, y, z, V, \Psi)\|.d\Psi \\ OF_{\text{right}}^{\mathcal{P}_{\text{avgf}}}(x, y, z, V) &= \frac{4}{\pi} \int_{-\pi/2}^0 \|OF(x, y, z, V, \Psi)\|.d\Psi \end{aligned} \quad (3.3)$$

The maximum optic flow pooling is noted \mathcal{P}_{max} and computed according to equation (3.4) for viewing angles $\Psi \in [0, \pi]$ and $\Psi \in [-\pi, 0]$.

$$\begin{aligned} OF_{\text{left}}^{\mathcal{P}_{\text{max}}}(x, y, z, V) &= \max_{\Psi \in [0, \pi]} \|OF(x, y, z, V, \Psi)\| \\ OF_{\text{right}}^{\mathcal{P}_{\text{max}}}(x, y, z, V) &= \max_{\Psi \in [-\pi, 0]} \|OF(x, y, z, V, \Psi)\| \end{aligned} \quad (3.4)$$

The maximum optic flow pooling in the frontal field of view is noted $\mathcal{P}_{\text{maxf}}$ and computed according to equation (3.5) for viewing angles $\Psi \in [0, \pi/2]$ and $\Psi \in [-\pi/2, 0]$.

$$\begin{aligned} OF_{\text{left}}^{\mathcal{P}_{\text{maxf}}}(x, y, z, V) &= \max_{\Psi \in [0, \pi/2]} \|OF(x, y, z, V, \Psi)\| \\ OF_{\text{right}}^{\mathcal{P}_{\text{maxf}}}(x, y, z, V) &= \max_{\Psi \in [-\pi/2, 0]} \|OF(x, y, z, V, \Psi)\| \end{aligned} \quad (3.5)$$

Ventral optic flow is modelled as the ratio between flight speed and vertical position:

$$OF_{\text{down}}(x, y, z, V) = \frac{V}{z} \quad (3.6)$$

3.2.4 Force field for flight control

The optic flow balancing behaviour of bees is here modelled as a force field. The force field $f : \mathcal{R}^4 \rightarrow \mathcal{R}^3$ defines three forces f_x , f_y and f_z for every agent state (x, y, z, V) .

$$\vec{f}(x, y, z, V) = \begin{bmatrix} f_x(x, y, z, V) \\ f_y(x, y, z, V) \\ f_z(x, y, z, V) \end{bmatrix} \quad (3.7)$$

The forces f_x , f_y and f_z respectively drive the agent along the x axis (i.e. controls flight speed), the y axis (i.e. controls the lateral position), and the z axis (i.e. controls the vertical position).

Force field for predictions of lateral position, vertical position, and flight speed. The predictions with uncoupled forward and vertical control (Fig. 3.4 and Fig. 3.5) are made using the force field presented in equation [3.8].

$$\begin{aligned} f_x &= OF_{\text{ref}} - \frac{OF_{\text{left}}^{\mathcal{P}_x} + OF_{\text{right}}^{\mathcal{P}_x}}{2} \\ f_y &= OF_{\text{right}}^{\mathcal{P}_y} - OF_{\text{left}}^{\mathcal{P}_y} \\ f_z &= OF_{\text{down}} - OF_{\text{ref}} \end{aligned} \quad (3.8)$$

where \mathcal{P}_x and \mathcal{P}_y are the optic flow pooling methods used for lateral control and speed control, $OF_{\text{left}}^{\mathcal{P}}$ and $OF_{\text{right}}^{\mathcal{P}}$ are the optic flow pooled using the pooling method \mathcal{P} on the left and right side, according to equations (3.2, 3.3, 3.4, 3.5), OF_{down} is the ventral optic flow computed according to equation [3.6], and OF_{ref} is a constant reference value. Note that this commonly accepted control scheme – although valid in a constrained environment like the one used in this study – would suffer from a scaling issue in an open, flat environment. Indeed, without objects on the sides, the lateral optic flow would be null and the agent would accelerate forward and upward infinitely, trying to match the reference OF_{ref} . In order to solve this inconsistency, our study may be extended to include viewing directions pointed slightly above and below the horizontal plane into the calculations of the lateral optic flow, in which case the ground would remain visible in a portion of the lateral visual field when flying in an open environment. Furthermore, bumblebees are likely to rely on a additional control mechanism involving the measurement of airspeed; this is, however, outside of the scope of our study.

Force field for predictions with coupled vertical and forward control. The predictions with coupled forward and vertical control (Fig. 3.7 and Fig. 3.6) are made using the force field presented in equation [3.9].

$$\begin{aligned}
 f_x &= OF_{\text{ref}} - \frac{OF_{\text{left}}^{\mathcal{P}_x} + OF_{\text{right}}^{\mathcal{P}_x}}{2} \\
 f_y &= OF_{\text{right}}^{\mathcal{P}_y} - OF_{\text{left}}^{\mathcal{P}_y} \\
 f_z^* &= OF_{\text{down}} - \frac{OF_{\text{left}}^{\mathcal{P}_z} + OF_{\text{right}}^{\mathcal{P}_z}}{2}
 \end{aligned} \tag{3.9}$$

where \mathcal{P}_x , \mathcal{P}_y and \mathcal{P}_z are the optic flow pooling methods used for lateral control, speed control, and vertical control, $OF_{\text{left}}^{\mathcal{P}}$ and $OF_{\text{right}}^{\mathcal{P}}$ are the optic flow pooled using the pooling method \mathcal{P} on the left and right side, according to equations (3.2, 3.3, 3.4, 3.5), OF_{down} is the ventral optic flow computed according to equation [3.6], and OF_{ref} is a constant reference value. For vertical control – unlike in the uncoupled case (equation [3.8]) – the ventral flow is compared to a non-constant reference value computed from lateral optic flow.

3.2.5 Prediction method

Lateral position y_{pred} , vertical position z_{pred} , and speed V_{pred} are predicted using an iterative gradient descent approach. As a first step, (y, z, V) are given initial values (y_0, z_0, V_0) . The initial values are found using a grid search with rough resolution in order to avoid local minima later in the gradient descent algorithm. The initial values are those where the average force applied to an agent while it is flying along the longitudinal axis of the corridor is the lowest.

$$\begin{pmatrix} V_0 \\ y_0 \\ z_0 \end{pmatrix} = \operatorname{argmin}_{y,z,V} \left(\left\| \int_{x_{\min}}^{x_{\max}} \vec{f}(x, y, z, V) \cdot dx \right\| \right) \tag{3.10}$$

At the k -th iteration, the current lateral position, vertical position and flight speed (y_k, z_k, V_k) define a straight trajectory flown at constant speed along the longitudinal axis x of the corridor. In order to know whether this trajectory is an equilibrium point – the predicted trajectory – or whether the agent will be pushed away from this trajectory by the applied forces, the force

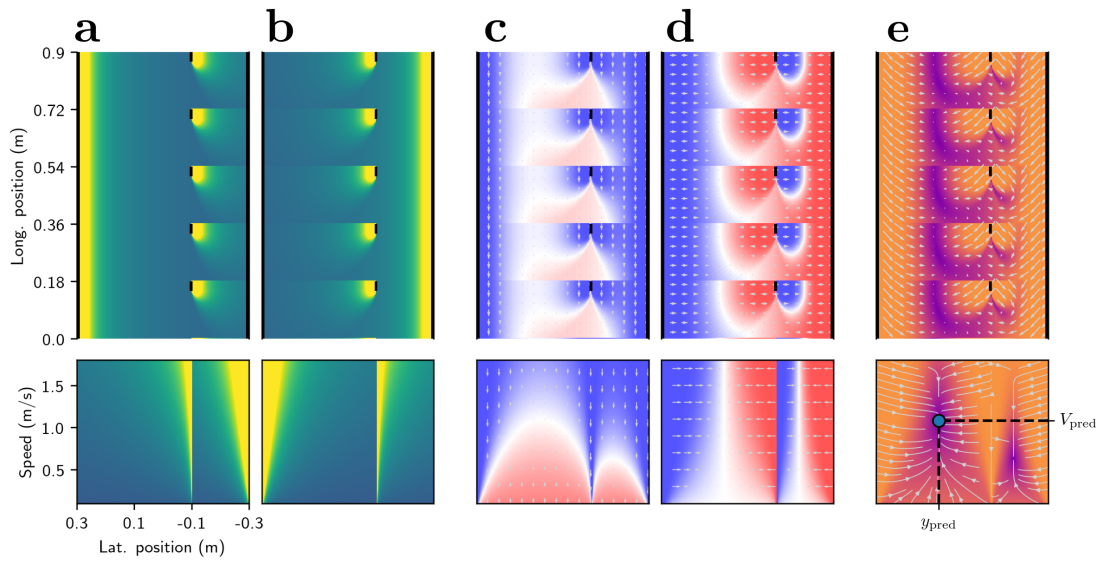


Figure 3.2 – Illustration of the prediction method for maxf optic flow pooling. **(a and b)**: Optic flow amplitude on the left side $OF_{\text{left}}^{\mathcal{P}_{\text{maxf}}}$ and on the right side $OF_{\text{right}}^{\mathcal{P}_{\text{maxf}}}$, computed according to equation [3.5]. Blue represents low optic flow amplitude, yellow represents large optic flow amplitude. **(c)**: Forward force f_x used to control flight speed and computed according to equation [3.8]. Blue represents negative f_x values (deceleration) and red represent positive f_x values (acceleration). **(d)**: Lateral force f_y used to control lateral position and computed according to equation [3.8]. Blue represents negative f_y values (rightward force) and red represent positive f_y values (leftward force). **(e)**: Combined force field \vec{f} . The white arrows represent the trajectories of an agent driven by the force field. The prediction is found by following the white arrow by gradient descent (see equation [3.12]).

field is evaluated and summed along the trajectory.

$$\begin{pmatrix} dV_k \\ dy_k \\ dz_k \end{pmatrix} = \int_{x_{\min}}^{x_{\max}} \vec{f}(x, y_k, z_k, V_k) \cdot dx \quad (3.11)$$

The result of this operation is an increment in flight speed dV , an increment in lateral position dy , and an increment in vertical position dz , that are used to iteratively update the predicted positions and speed by gradient descent.

$$\begin{pmatrix} V_{k+1} \\ y_{k+1} \\ z_{k+1} \end{pmatrix} = \begin{pmatrix} V_k \\ y_k \\ z_k \end{pmatrix} + \begin{pmatrix} dV_k \\ dy_k \\ dz_k \end{pmatrix} \quad (3.12)$$

The iterative process described in equation [3.11] and equation [3.12] is repeated until a minimum number of iterations are performed and the increments dV , dy and dz are below a small threshold (10^{-6} in our case).

3.2.6 Prediction error

The predictions are evaluated by comparing them to the recorded bee trajectories to obtain the prediction error e

$$e = \sqrt{e_x^2 + e_y^2 + e_z^2} \quad (3.13)$$

which is computed from the error on flight speed e_x , lateral position e_y and vertical position e_z

$$\begin{aligned} e_x &= \sum_{\mathcal{C}_{x|x}} \frac{1}{N_C} \frac{V_{\text{pred}} - \mu_V}{\sigma_V} \\ e_y &= \sum_{\mathcal{C}_{x|x}} \frac{1}{N_C} \frac{y_{\text{pred}} - \mu_y}{\sigma_y} \\ e_z &= \sum_{\mathcal{C}_{x|x}} \frac{1}{N_C} \frac{z_{\text{pred}} - \mu_z}{\sigma_z} \end{aligned} \quad (3.14)$$

where $\mathcal{C}_{x|x}$ indicates each one of the N_C tested conditions. V_{pred} , y_{pred} , and z_{pred} are the predicted flight speed, lateral position and vertical position. μ_V , μ_y and μ_z are the mean of the measured flight speed, lateral position and vertical position for each condition. σ_V , σ_y and σ_z are the standard deviations of the measured flight speed, lateral position and vertical

position for each condition.

In equation [3.14], the error between predicted values and mean of the measured values is divided by the standard deviation of the measured values. This allows comparing errors on different axis even though they have different scales and units. In addition, dividing the prediction errors by the standard deviation of the measured data puts more emphasis on predictions in cases where bees behaved in a consistent manner (i.e. low standard deviation) compared to the cases where bees had more variable behaviour (i.e. large standard deviation).

3.3 Results and discussion

First, we recorded the three-dimensional position of bumblebees flying along an experimental corridor in which we varied the density of obstacles placed along two lines parallel to the main axis of the corridor at a distance of 0.2 m from each wall. The experimental conditions are named according to the format $\mathcal{C}_{\alpha|\beta}$ where α and β represent the percentage of the corridor length occupied by obstacles on the left and right side, respectively, of a bee flying towards a feeder. For example, the corridor $\mathcal{C}_{0|16}$ has no obstacles on the left side, and obstacles covering 16% of the tunnel length on the right side (Fig. 3.1), while in the corridor $\mathcal{C}_{33|33}$ obstacles occupy 33% of the corridor length on both sides (Fig. 3.3a).

3.3.1 The effect of obstacle density on lateral position

In the conditions without obstacles, $\mathcal{C}_{0|0}$ and $\mathcal{C}_{0|100}$ (Fig. 3.3a), bumblebees maintain an equal distance to both walls. With obstacles on one side (conditions $\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$, Table 3.1) they instead fly between the obstacles and the wall, close to the safest position of $y = 0.1$ m (dashed grey line, Fig. 3.3) but as obstacle density decreases, they fly slightly closer to the obstacles (Fig. 3.3c).

Flights were also centred in corridors with symmetrically distributed obstacles ($\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$), with the lateral position being less variable (lower standard deviations) than in the corridor without obstacles $\mathcal{C}_{0|0}$. This is likely due to the factor of 3 reduction in effective corridor width (0.2 m between the obstacles for $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$, compared to 0.6 m between the walls in $\mathcal{C}_{0|0}$), although the standard deviation decreased by a factor of 3.8 to 4.75 (Table 3.1). Similarly, even though the maximum width is higher in $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$ than in $\mathcal{C}_{0|100}$ (0.6 m compared to 0.4 m), the standard deviation of the lateral position is lower in conditions $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$ than in conditions $\mathcal{C}_{0|100}$. This result suggests that bumblebees respond to the presence of obstacles by increasing the precision with which they control lateral position.

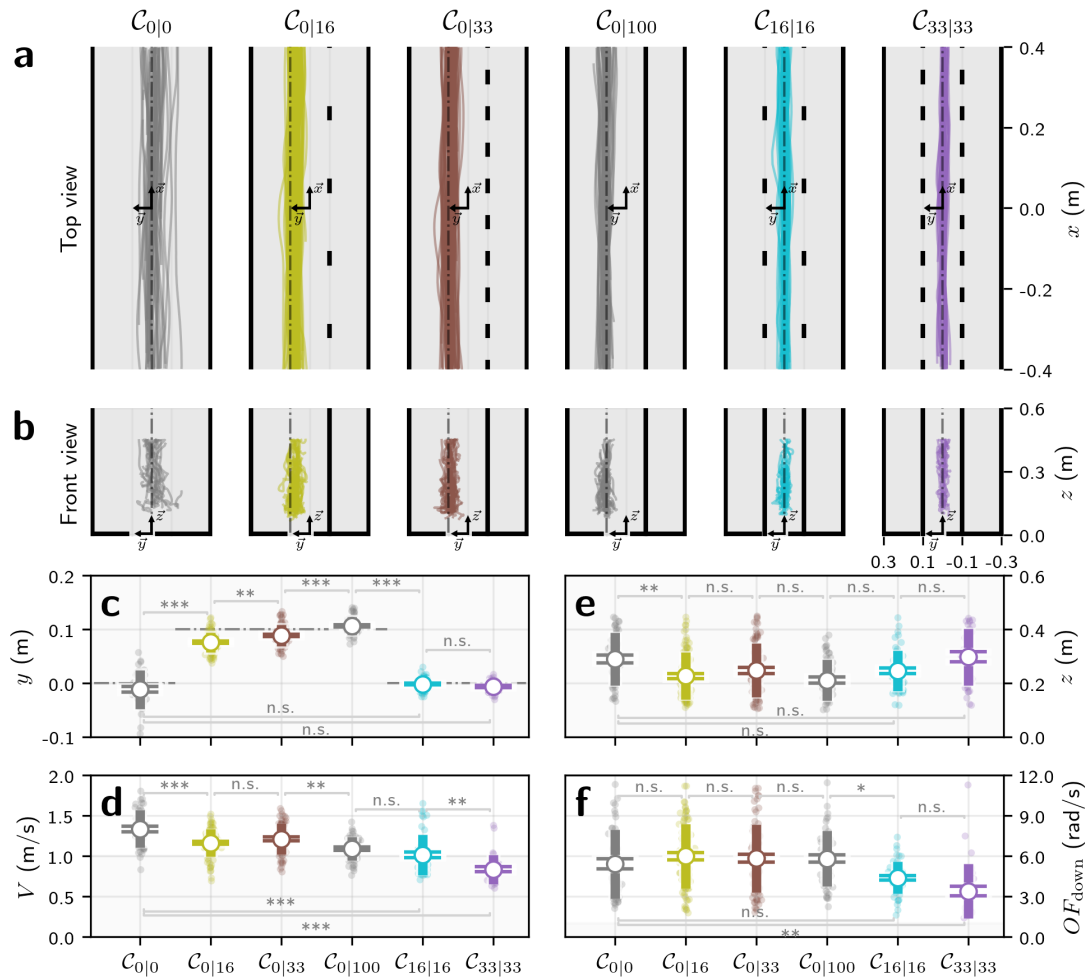


Figure 3.3 – Effect of obstacle density on flight control in bumblebees.

In condition $C_{0|0}$, there are no obstacles. In conditions $C_{0|16}$ and $C_{0|33}$, the obstacles are only on one side of the corridor and cover 16% and 33% of its length, respectively. In $C_{0|100}$, the obstacles and the wall on the opposite side form a narrower corridor centred on lateral position $y = 0.1$ m. In conditions $C_{16|16}$ and $C_{33|33}$, the obstacles are on both sides and cover 16% and 33% of the corridor length, respectively. **(a)**: Top view of the experimental corridors and recorded trajectories for all conditions. **(b)**: Front view of the experimental corridors and recorded trajectories for all conditions. **(c)**: Measured lateral positions y , expressed in m. **(d)**: Measured flight speeds V , expressed in m/s. **(e)**: Measured vertical positions z , expressed in m. **(f)**: Ventral flow OF_{down} , expressed in rad/s, and computed as $OF_{\text{down}} = V/z$ from the measured vertical positions and flight speeds. **(c, d, e, f)**: Measurements are displayed as coloured open circles. The horizontal lines on the error bars denote the standard error of the mean. The uncapped bars denote the standard deviation. The statistical significance of the difference between measured data across two corridor conditions is tested using post-hoc Tukey HSD tests, with the null hypothesis that the mean recorded data is equal for both conditions. The horizontal grey lines indicate the p-value of the post-hoc Tukey HSD tests (***) when $p \leq 0.001$, ** when $0.001 < p \leq 0.01$, * when $0.01 < p \leq 0.05$ and *n.s.* when $0.05 < p$).

Chapter 3. The role of optic flow pooling in insect visual flight control

		Measurements				Tukey HSD test					
		μ	σ	N	n	$\mathcal{C}_{0 0}$	$\mathcal{C}_{0 16}$	$\mathcal{C}_{0 33}$	$\mathcal{C}_{0 100}$	$\mathcal{C}_{16 16}$	$\mathcal{C}_{33 33}$
Lateral Position	$\mathcal{C}_{0 0}$	-0.012	0.036	46	26		***	***	***	n.s.	n.s.
	$\mathcal{C}_{0 16}$	0.076	0.018	87	35	0.001		**	***	***	***
	$\mathcal{C}_{0 33}$	0.088	0.020	73	31	0.001	0.002		***	***	***
	$\mathcal{C}_{0 100}$	0.107	0.017	50	16	0.001	0.001	0.001		***	***
	$\mathcal{C}_{16 16}$	-0.002	0.012	49	25	0.163	0.001	0.001	0.001		n.s.
	$\mathcal{C}_{33 33}$	-0.006	0.011	32	19	0.805	0.001	0.001	0.001	0.900	
Flight Speed	$\mathcal{C}_{0 0}$	1.337	0.235	46	26		***	**	***	***	***
	$\mathcal{C}_{0 16}$	1.164	0.171	87	35	0.001		n.s.	n.s.	***	***
	$\mathcal{C}_{0 33}$	1.212	0.189	73	31	0.010	0.605		**	***	***
	$\mathcal{C}_{0 100}$	1.091	0.147	50	16	0.001	0.286	0.010		n.s.	***
	$\mathcal{C}_{16 16}$	1.013	0.249	49	25	0.001	0.001	0.001	0.364		**
	$\mathcal{C}_{33 33}$	0.836	0.181	32	19	0.001	0.001	0.001	0.001	0.001	
Vertical Position	$\mathcal{C}_{0 0}$	0.289	0.098	46	26		**	n.s.	***	n.s.	n.s.
	$\mathcal{C}_{0 16}$	0.226	0.088	87	35	0.002		n.s.	n.s.	n.s.	**
	$\mathcal{C}_{0 33}$	0.247	0.099	73	31	0.136	0.670		n.s.	n.s.	n.s.
	$\mathcal{C}_{0 100}$	0.211	0.076	50	16	0.001	0.900	0.251		n.s.	***
	$\mathcal{C}_{16 16}$	0.245	0.075	49	25	0.177	0.827	0.900	0.412		n.s.
	$\mathcal{C}_{33 33}$	0.297	0.106	32	19	0.900	0.003	0.097	0.001	0.124	

Table 3.1 – Summary of the experimental measurements and statistical significance of the differences measured between pairs of conditions.

(Measurements): For each corridor condition ($\mathcal{C}_{0|0}$, $\mathcal{C}_{0|16}$, $\mathcal{C}_{0|33}$, $\mathcal{C}_{0|100}$, $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$), n bees were recorded while they performed N flights across the corridors. For each flight, the lateral position (in m), vertical position (in m), and flight speed (in m/s) are averaged. μ and σ are respectively the mean and standard deviation across the N flights. **(Tukey HSD test):** The p-values of post-hoc Tukey HSD tests between values measured in each pair of corridor conditions. The p-values are shown in the entries below the diagonal, and their significance levels are indicated in the corresponding above-diagonal entries. Cases where the null-hypothesis that lateral position, vertical position, or flight speed have equal means cannot be rejected are indicated by "n.s.". The symbols indicate the p-value of the tests: *** when $p \leq 0.001$, ** when $0.001 < p \leq 0.01$, * when $0.01 < p \leq 0.05$ and n.s. when $0.05 < p$.

In $\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$, bumblebees do not fly at an equal distance between the obstacles and the opposite wall, but slightly closer to the obstacles ($p \leq 0.001$, Table 3.1). As the density of obstacles increases on one side (0% in $\mathcal{C}_{0|0}$, 16% in $\mathcal{C}_{0|16}$, and 33% in $\mathcal{C}_{0|33}$), bumblebees increase their distance from them ($p \leq 0.01$, Table 3.1).

3.3.2 The effect of obstacle density on flight speed

In the conditions without obstacles ($\mathcal{C}_{0|0}$ and $\mathcal{C}_{0|100}$), bumblebees adjust their flight speed to the corridor width (Fig. 3.3d). Indeed, speed control in flying insects is often modelled as maintaining the magnitude of optic flow – that is, the ratio between flight speed (V) and the distance to nearby surfaces (D) – at a predefined value (for a review see [124]), which has been found experimentally to lie between 3.0 rad/s and 6.0 rad/s [81, 7, 4, 117] for honeybees and bumblebees. In the 0.4 m wide corridor ($\mathcal{C}_{0|100}$), the average flight speed was 1.1 m/s, which would generate a lateral optic flow of magnitude 5.5 rad/s. In the 0.6 m wide corridor ($\mathcal{C}_{0|0}$), the average flight speed increased to 1.3 m/s, generating a lateral optic flow of magnitude 4.3 rad/s. This represents a speed increase of 20% but this does not match the proportional increase of corridor width, which was 50% and suggests that lateral optic flow is not the only source of information used for flight speed control and is consistent with the findings of previous studies in both bumblebees [82] and honeybees [4, 100].

In the presence of obstacles, bumblebees decrease their flight speed from 1.3 m/s in $\mathcal{C}_{0|0}$, which is an empty corridor, to 1.0 m/s in $\mathcal{C}_{16|16}$, where obstacles cover 16% of the corridor length on each side ($p \leq 0.001$, Table 3.1). Flight speed decreases further to 0.84 m/s when the density of obstacles increases to 33% on each side in $\mathcal{C}_{33|33}$ ($p \leq 0.01$, Table 3.1). With obstacles on only one side of the corridor, speed decreases from 1.3 m/s in $\mathcal{C}_{0|0}$ to 1.16 m/s in $\mathcal{C}_{0|16}$ ($p \leq 0.001$, Table 3.1). Flight speed tends to increase with increasing obstacle density from $\mathcal{C}_{0|16}$ to $\mathcal{C}_{0|33}$, although not significantly so ($p = 0.605$, Table 3.1). Finally, speed decreases from 1.21 m/s in $\mathcal{C}_{0|33}$, where obstacles cover 33% of the corridor length on one side, to 1.09 m/s in $\mathcal{C}_{0|100}$, where the obstacles form a complete wall on one side ($p \leq 0.01$, Table 3.1).

Interestingly, bumblebees fly slower when obstacles are present on both sides than when obstacles are present on only one side. They decrease their flight speed from 1.16 m/s in $\mathcal{C}_{0|16}$ to 1.01 m/s in $\mathcal{C}_{16|16}$ ($p \leq 0.001$, Table 3.1) and from 1.21 m/s in $\mathcal{C}_{0|33}$ to 0.84 m/s in $\mathcal{C}_{33|33}$ ($p \leq 0.001$, Table 3.1). This indicates that flight speed is controlled using optic flow on both sides and that it is affected by the presence and density of obstacles.

3.3.3 The effect of obstacle density on vertical position

In the absence of obstacles, bumblebees maintain the same distance from the floor as they do from the wall: when the half-width of the corridor is 0.3 m ($C_{0|0}$) the average vertical position is 0.29 m and, similarly, when the half-width of the corridor is 0.2 m, the average vertical position is 0.21 m. In other words, in corridors that do not contain obstacles, the magnitude of the lateral optic flow is held equal to the magnitude of the ventral flow.

In all conditions, the vertical position is quite variable with relatively high standard deviations (Fig. 3.3e) and there are few cases where the differences between the conditions were significant (Table 3.1). Interestingly, ventral optic flow (calculated by dividing the measured flight speed by the measured vertical position, Fig. 3.3f) does not vary between conditions $C_{0|0}$, $C_{0|16}$, $C_{0|33}$, and $C_{0|100}$, but decreases significantly in $C_{16|16}$ and $C_{33|33}$. This is a surprising result because the optic flow profile generated in the ventral visual field is the same for all conditions, so ventral flow should not be affected by the density of obstacles in the lateral visual field. We would expect bumblebees to fly significantly lower in $C_{16|16}$ and $C_{33|33}$ in order to compensate for the reduced flight speed and to maintain the ventral optic flow at a constant value. However, this is not what we observe (Fig. 3.3e-f) and suggests that vertical control might not be driven only by ventral optic flow, as previously modelled [125, 102, 99, 5] but that it might be mediated by a combination of ventral and lateral optic flow.

3.3.4 Predicting the effect of obstacle density and optic flow pooling on flight control

The change in lateral position, flight speed, and vertical position in response to the density of nearby obstacles is likely a reflection of the way in which bumblebees pool optic flow from their panoramic field of view to control flight. Are they using a fixed spatial integration of optic flow across their entire visual field or are they selectively reacting to the nearby obstacles? To answer this question, we predict the lateral position, flight speed and vertical position in each of our experimental corridors for different methods of optic flow integration from different parts of the visual field. We then assess how well the results of each method agree with the experimental data from bumblebees to determine which method best explains the bees' responses.

3.3.5 Predicted lateral position

Our predictions of the lateral position are based on the optic flow balancing control strategy in which a sideways force is applied based on the difference between the optic flow on the left

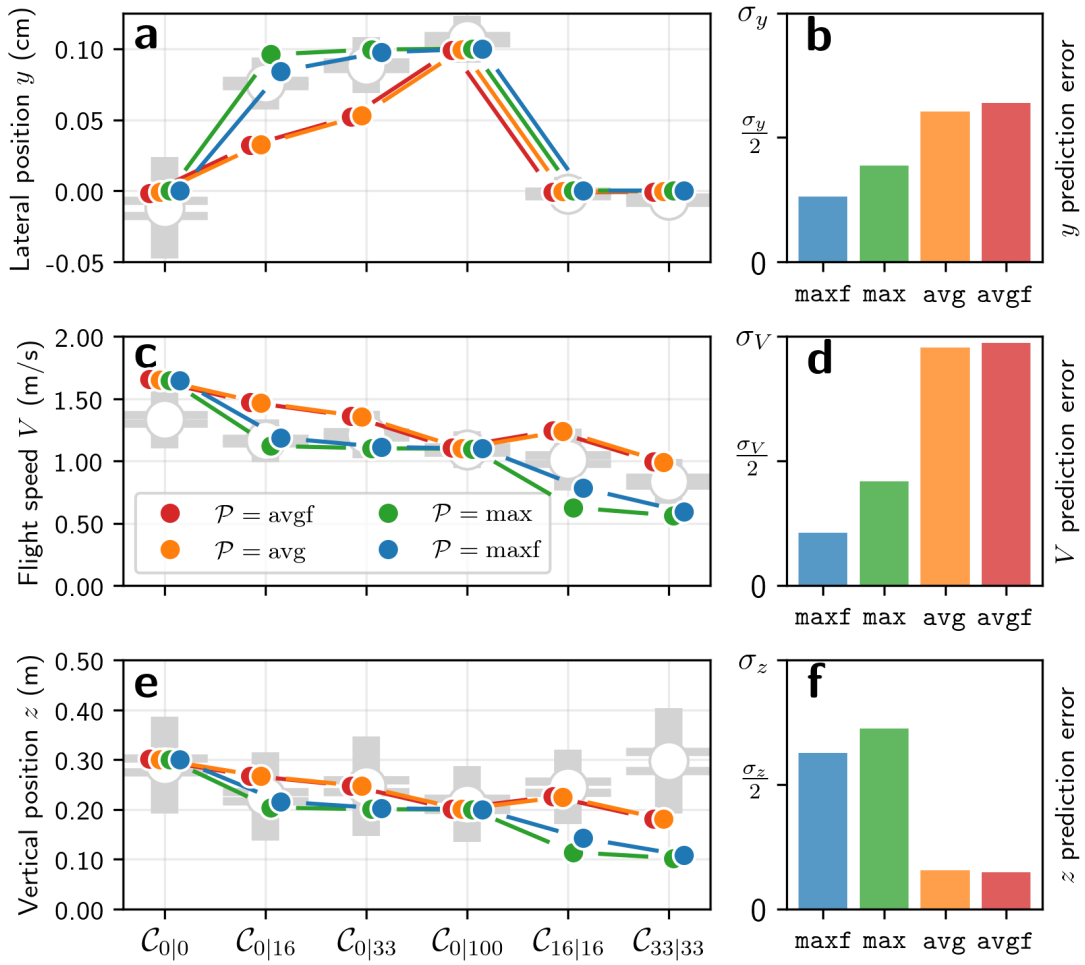


Figure 3.4 – Predicted lateral position, flight speed and vertical position.

The experimental measurements of bumblebee lateral position (y), flight speed (V), and vertical position (z) are compared to predicted values. Predictions are made according to the control method described in equation [3.8], with the same optic flow pooling \mathcal{P} on all axes ($\mathcal{P} = \mathcal{P}_x = \mathcal{P}_y$). Predictions are made for four optic flow pooling methods avg, avgf, max, and maxf described in equations (3.2, 3.3, 3.4, 3.5). The best prediction for lateral position and flight speed are made with maxf pooling, while the best predictions for vertical position are made with avg and avgf pooling. (a): Measured lateral position and predicted lateral position for each experimental condition. (b): Average prediction error for lateral position. (c): Measured flight speed and predicted flight speed for each experimental condition. (d): Average prediction error for flight speed. (e): Measured vertical position and predicted vertical position for each experimental condition. (f): Average prediction error for vertical position. (a, c, e): Measurements are displayed as grey open circles. The horizontal lines on the error bars denote standard error of the mean, the uncapped bars denote the standard deviation. Predictions are displayed as solid coloured circles. Note that predictions are shifted horizontally for clarity. (b, d, f): Each prediction is compared with experimental data as shown in equation [3.14]. The prediction errors are averaged over conditions $\mathcal{C}_{0|0}$, $\mathcal{C}_{0|16}$, $\mathcal{C}_{0|33}$, $\mathcal{C}_{0|100}$, $\mathcal{C}_{16|16}$, and $\mathcal{C}_{33|33}$.

Chapter 3. The role of optic flow pooling in insect visual flight control

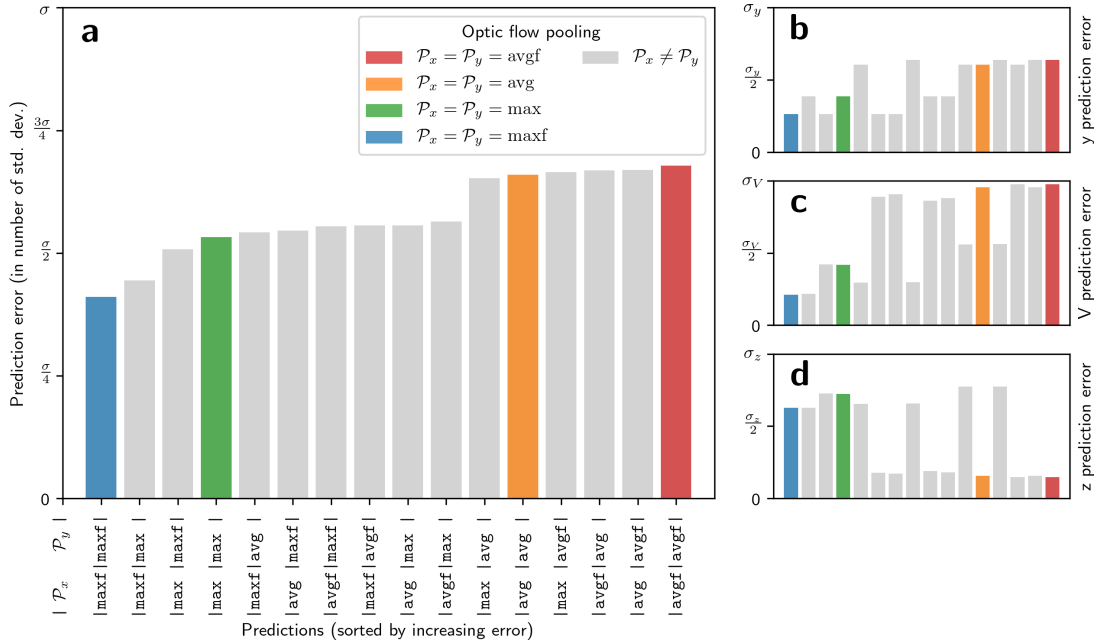


Figure 3.5 – Prediction errors of lateral position, vertical position and speed. Average prediction errors for different optic flow pooling \mathcal{P}_x and \mathcal{P}_y . The predictions of flight speed and lateral position are made using different pooling \mathcal{P}^x and \mathcal{P}^y respectively for the control of flight speed and lateral position. The predictions are made according to the control method shown in equation (3.8). Each prediction is compared with experimental data. The resulting error is scaled according to the standard deviation of the experimental data as shown in equation (3.14) to allow comparison between values predicted on different axes and units. The scaled errors are then averaged over conditions $\mathcal{C}_{0|0}$, $\mathcal{C}_{0|16}$, $\mathcal{C}_{0|33}$, $\mathcal{C}_{0|100}$, $\mathcal{C}_{16|16}$, and $\mathcal{C}_{33|33}$ computed according to equation (3.13). (a): Average prediction error on all axes. (b): Average prediction error for flight speed. (c): Average prediction error for lateral position. (d): Average prediction error for vertical position.

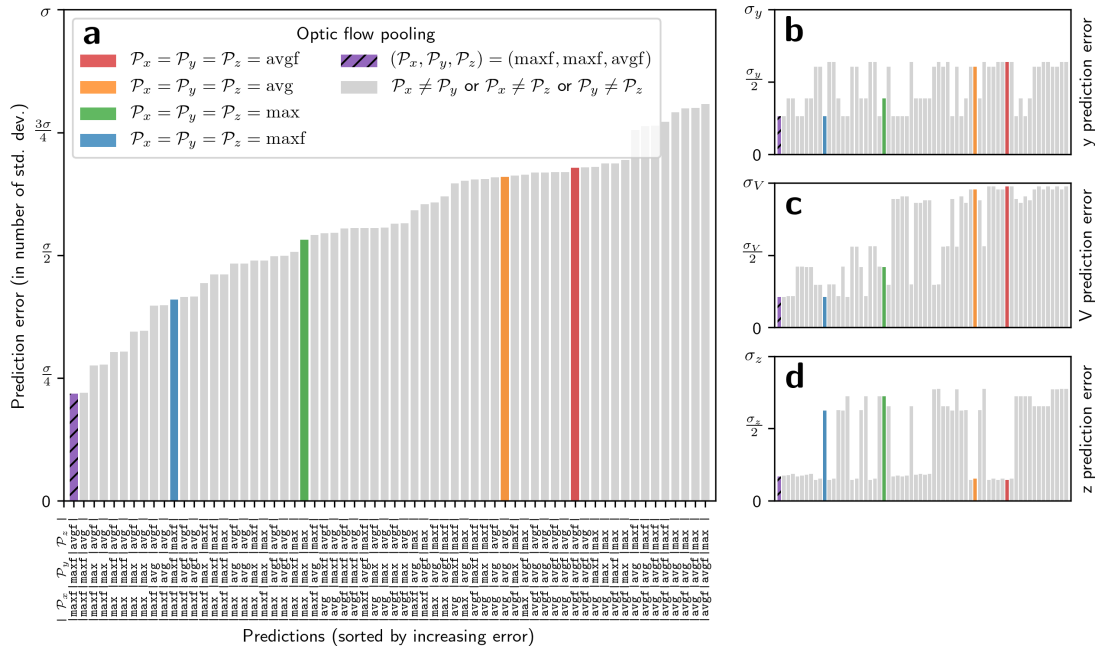


Figure 3.6 – Prediction error when combining multiple optic flow pooling methods, with coupled forward and vertical control.

Average prediction errors when different optic flow pooling methods are used to control different aspects of flight. All combinations of optic flow pooling avg, avgf, max, and maxf are tested for the control of lateral position, flight speed and vertical position. For each combination, the prediction error on each axis is computed according to equation [3.14], then combined into a single error according to equation [3.13]. The predictions are made according to the control method shown in equation [3.9], where the reference optic flow used for vertical control is not constant but is computed from the lateral optic flow. The predictions are made using different pooling \mathcal{P}^x , \mathcal{P}^y , and \mathcal{P}^z for the control of flight speed, lateral position, and vertical position, respectively. Contrary to the uncoupled case (see Fig. 3.5), the best prediction is achieved when combining different optic flow pooling methods. With maximum optic flow pooling in the frontal visual field for forward and lateral control ($\mathcal{P}_x = \mathcal{P}_y = \text{maxf}$) and average optic flow pooling in the frontal visual field for vertical control ($\mathcal{P}_z = \text{avgf}$), the prediction error is the lowest, and is 50% smaller than with $\mathcal{P}_x = \mathcal{P}_y = \mathcal{P}_z = \text{maxf}$. (a): Average prediction error on all axes. (b): Average prediction error for flight speed. (c): Average prediction error for lateral position. (d): Average prediction error for vertical position.

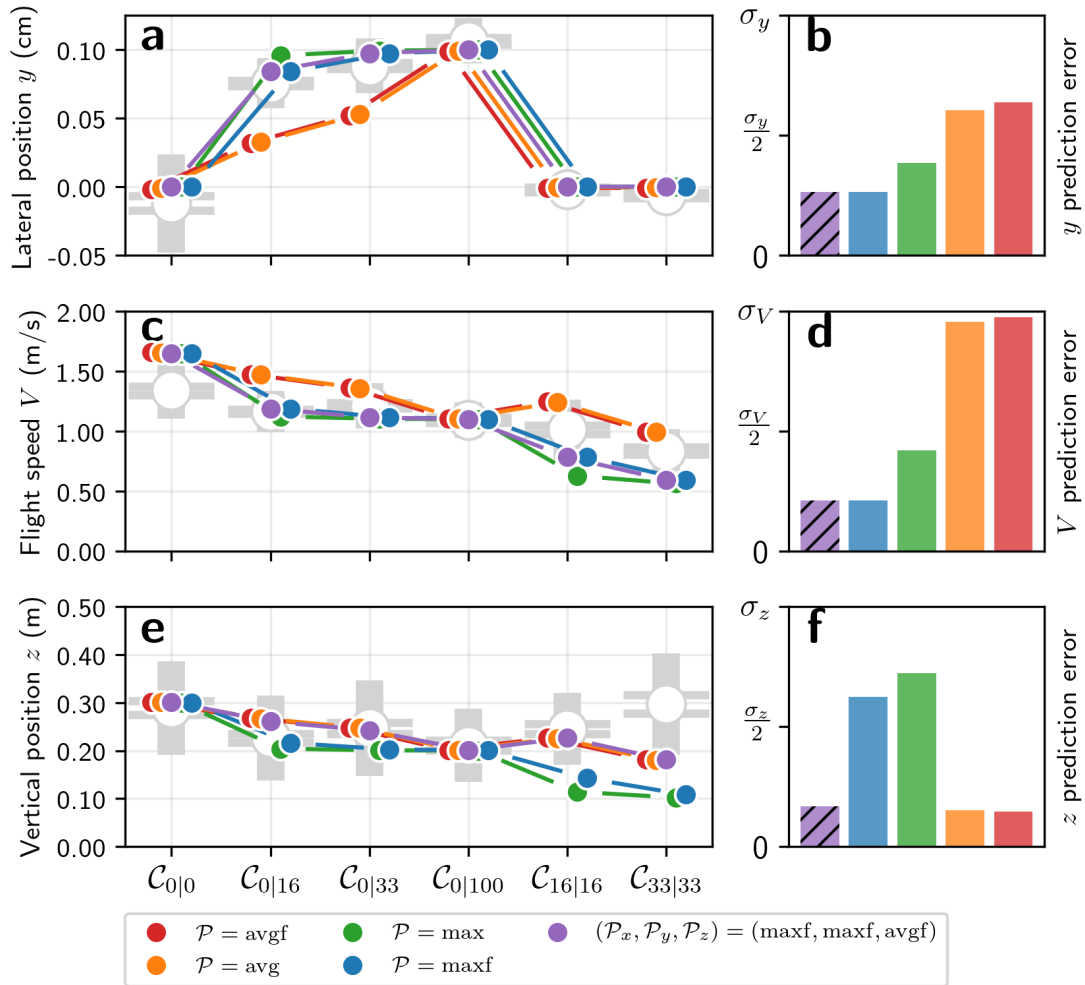


Figure 3.7 – Predicted lateral position, flight speed and vertical position with coupled forward and vertical control. The experimental measurements of bumblebee lateral position (y), flight speed (V), and vertical position (z) are compared to predicted values. Predictions are made according to the control method described in equation [3.9], where the control of flight speed and vertical position are coupled, and different pooling methods can be combined. For more clarity, all the combinations of optic flow pooling are not presented here. The same pooling is used on all control axes for all presented predictions except one (shown in dashed purple), which yields the lowest prediction error. (a): Measured lateral position and predicted lateral position for each tested conditions. (b): Average prediction error for lateral position. (c): Measured flight speed and predicted flight speed for each tested conditions. (d): Average prediction error for flight speed. (e): Measured vertical position and predicted vertical position for each tested conditions. (f): Average prediction error for vertical position. (a, c, e): Measurements are displayed as grey open circles. Similarly to Fig. 3.3, the horizontal lines on the error bars denote standard error of the mean. The uncapped bars denote the standard deviation. Predictions are displayed as solid coloured circles. Note that predictions are shifted horizontally for better presentation. (b, d, f): Each prediction is compared with experimental data as shown in equation [3.14].

and right sides [124] (for details, see equation [3.8]). Optic flow on the left and right side are pooled according to four of the most biologically plausible methods: average optic flow across lateral and fronto-lateral visual field (noted respectively *avg* and *avgf*), and maximum optic flow in lateral and fronto-lateral visual field (noted respectively *max* and *maxf*).

In the symmetric conditions ($\mathcal{C}_{0|0}$, $\mathcal{C}_{0|100}$, $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$), all predictions match the measured lateral positions (Fig. 3.4a), because the predicted lateral position for these conditions does not depend on the optic flow pooling method and reveals little about the specific optic flow pooling strategy being used. In the asymmetric conditions that have obstacles only on one side ($\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$), the predicted lateral position does vary with the optic flow integration method. It is interesting to note that the predicted lateral positions for average optic flow (*avg* and *avgf*) do not vary with changes in the field of view (Fig. 3.4a). Also, the predictions made using maximum rate of optic flow (*max* and *maxf*) have a greater distance to obstacles than those made using averaged optic flow (*avg* and *avgf*). This is not surprising because with the maximum optic flow the agent selectively reacts to the nearby obstacles, while averaging causes the optic flow from the nearby obstacles to 'blend' into the background, reducing their influence on lateral position. The predictions made using the maximum optic flow in the frontal visual field lie the closest to the safest position (at $y = 0.1$ m, i.e. the lateral position at equal distance from the line of obstacles and the opposite wall), and yield lower error than the predictions made using average optic flow when compared to the bumblebee data (Fig. 3.4b). Optic flow pooling *avg* and *avgf* generate lateral positions much closer to the obstacles – and thus much less safe trajectories – than those performed by insects. This suggests that bumblebees use the maximum optic flow in the frontal visual field to control their lateral position, which is the optimal approach because i) the frontal visual field is where incoming obstacles are the most likely to occur and ii) selecting the maximum optic flow will ensure that the bees selectively react to the closest obstacles in the visual field.

3.3.6 Predicted flight speed

Our predictions of flight speed are based on an optic flow regulation strategy in which a longitudinal force derived from the difference between lateral optic flow and a reference optic flow value is applied [124, 81] (for details, see equation [3.8]). The agent accelerates if the lateral optic flow is lower than the reference optic flow, it decelerates if the lateral optic flow is higher than the reference optic flow, and remains at a constant speed when they are equal. The lateral optic flow is calculated by pooling optic flow on the left and right sides, then taking the mean of the two resulting signals. For the reference optic flow, we use a value of 5.5 rad/s, which is close to the values obtained in previous studies [81, 7, 4, 117], and is close to the

Chapter 3. The role of optic flow pooling in insect visual flight control

magnitude of lateral optic flow experienced by bumblebees in $\mathcal{C}_{0|100}$ (Fig. 3.3d, Table 3.1).

In conditions that do not contain obstacles ($\mathcal{C}_{0|0}$ and $\mathcal{C}_{0|100}$), the same speed is predicted with all pooling methods (Fig. 3.4c) and explains why data from simple corridors alone is insufficient for testing hypotheses about how insects pool optic flow for flight control. The predicted speed in $\mathcal{C}_{0|100}$ (1.10 m/s) is close to the measured speed (1.09 m/s), which is not surprising given that the reference optic flow value lies close to the optic flow value known to be used by bumblebees. However, the predicted speed for $\mathcal{C}_{0|0}$ is 1.65 m/s, which is significantly higher than the measured value of 1.34 m/s ($p \leq 0.001$). This discrepancy between measured and predicted flight speed in $\mathcal{C}_{0|0}$ may be explained by additional sensory feedback – such as airspeed – which is likely combined with optic flow for speed control [129]. However, there is currently no existing model for how airspeed and optic flow are combined to control flight speed in freely-flying insects.

The most interesting conditions for studying the effect of optic flow pooling on flight speed are those in which obstacles are present: $\mathcal{C}_{0|16}$, $\mathcal{C}_{0|33}$, $\mathcal{C}_{16|16}$, and $\mathcal{C}_{33|33}$. In these conditions, average pooling *avg* and *avgf* generates the highest predicted speeds because maximum pooling *max* and *maxf* (Fig. 3.4c) selects optic flow generated by the closest objects, leading to a reduced flight speed.

The variation of predicted flight speed with varying obstacle density suggests that, as with lateral position control, flight speed is regulated using maximum optic flow pooling. The flight speed predicted using maximum optic flow shows very little variation between $\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$ (Fig. 3.4c), which is also the case for the measured flight speeds (Table 3.1). On the contrary, the flight speed predicted using average optic flow shows a steeper decrease with increasing obstacle density in $\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$. Furthermore, the predicted flight speed with average pooling is higher in $\mathcal{C}_{16|16}$ than in $\mathcal{C}_{0|100}$, unlike measured flight speeds which show a non-significant decrease in flight speed (Table 3.1). The proximity of obstacles in $\mathcal{C}_{16|16}$ is higher than in $\mathcal{C}_{0|100}$ and it would thus be more sensible to reduce flight speed in this condition, as predicted by maximum pooling.

The predicted speeds lie closer to the measured speeds with maximum pooling than with average pooling (Fig. 3.4c). The error with *maxf* pooling is the lowest and is more than 4 times lower than with *avgf* pooling (Fig. 3.4d). These results suggest that – similar to lateral control – bees use maximum optic flow pooling in the frontal visual field to control their speed. This makes sense from a biological point of view because it selects the visual motion generated by the closest obstacles, which represent the main collision threats. Also, incoming obstacles are more likely to occur in the frontal visual field, which is coherent with the fact that predictions are more accurate with *maxf* pooling than with *max* pooling (Fig. 3.4d).

3.3.7 Predicted vertical position

Our predictions of vertical position are based on an optic flow regulation strategy in which a vertical force is applied based on the difference between the ventral optic flow and a reference optic flow value obtained from honeybees [99] (similar data does not exist for bumblebees; for details, see equation [3.8]). Note that honeybees use the dorsal flow for controlling vertical position when they are closer to the ceiling than to the ground and ventral flow when flying closer to the ground [101], we thus consider only the ventral flow in our predictions because the ceiling is located approximately 2 m above the 0.6 m high experimental tunnel. In our test environments, there are no obstacles between the agent and the floor; the ventral optic flow is thus independent of the elevation angle and all optic flow pooling methods provide the same result. We, therefore, modelled ventral optic flow as the ratio between the flight speed and the vertical position: $OF_{\text{down}} = V/z$, and do not apply pooling.

In the conditions that do not contain obstacles ($\mathcal{C}_{0|0}$ and $\mathcal{C}_{0|100}$), the predicted vertical positions match the measurements (Fig. 3.4e) and are equal to the corridor half-widths (0.2 m in $\mathcal{C}_{0|100}$ and 0.3 m in $\mathcal{C}_{0|0}$). This is because flight speed is regulated so that the mean of the pooled optic flow on left and right sides is equal to the reference optic flow value. In obstacle-free corridors, the agent tends to fly at an equal distance from both walls, meaning that pooled optic flow on the left and right sides take the same value, equal to the reference value. Finally, as the vertical position is regulated so that ventral optic flow is equal to the same reference value, the vertical position is equal to the distance between the agent and the walls.

Although no pooling is performed on ventral flow, the predicted vertical position is affected by flight speed, which in turn is affected by the presence of obstacles and thus also the pooling method used. The predicted vertical position is lower when maximum pooling is used on the lateral optic flow than when average pooling is used (Fig. 3.4e) because the agent flies slower and must reduce its height to maintain ventral optic flow at the reference value.

It is interesting to note that in $\mathcal{C}_{16|16}$ and $\mathcal{C}_{33|33}$ – i.e. with obstacles on both sides – the predicted vertical positions with max pooling are close to 0.1 m, which is half the distance between the rows of obstacles. Similarly, in $\mathcal{C}_{0|16}$ and $\mathcal{C}_{0|33}$ – i.e. with obstacles on one side only – the predicted vertical positions with max pooling are close to 0.2 m. Thus, when it comes to vertical control with maximum pooling, the agent acts as if the row of obstacles were a wall, which is not consistent with the behaviour of bumblebees (Fig. 3.3d-e).

The predictions that best match the experimental data are with average pooling, which yield prediction errors (Fig. 3.4f) that are approximately 5 times lower than with maximum pooling, suggesting that bees use average pooling of lateral optic flow to control their vertical position.

3.3.8 Combining multiple optic flow pooling methods for different aspects of flight control

Our model predictions suggest that bumblebees use maximum optic flow pooling in the frontal visual field to control both their lateral position and flight speed but that vertical position is controlled by average optic flow pooling in the lateral visual field. An agent using only maxf would correctly replicate bumblebee lateral position and flight speed, but would not fly at the same vertical position and an agent using only average pooling would correctly replicate bumblebee vertical position, but would not fly at the same lateral position and flight speed. How can we reconcile this contradiction?

Given the parallel nature of neural systems, the same piece of information can be processed by several circuits in the brain, each circuit implementing a different function. For example, the optomotor response and centring behaviour are known to be mediated by two distinct movement detecting pathways in the honeybee visual system [121]. Thus, optic flow from a wide field of view may be processed (or pooled) several times in parallel according to different functions – like average and maximum pooling – and across different visual fields before it is used to control different aspects of flight.

To test this hypothesis, we generated predictions using different optic flow pooling methods in parallel. Namely, in the formulation of the three forces driving the control of lateral position, flight speed and vertical position (equation [3.8]), we allowed different pooling methods to be used for each axis. While we applied the same control strategies for lateral position and flight speed as before, we modified the control strategy for vertical position. Instead of regulating the ventral optic flow so that it is equal to a fixed reference value, we regulate it according to a value generated from the lateral optic flow (and therefore coupled to flight speed, see equation [3.9]).

We tested all combinations of optic flow pooling ($\mathcal{P}_x, \mathcal{P}_y, \mathcal{P}_z$) and compared the prediction accuracy. While using multiple optic flow pooling methods did not improve the accuracy of our predictions when the control of speed and vertical position were uncoupled (equation [3.8] and Fig. 3.5), the predictions were more accurate when the control of flight speed and vertical position were coupled (equation [3.9] and Fig. 3.6). Interestingly, the best prediction is achieved with maximum optic flow pooling in the frontal visual field for speed and lateral control ($\mathcal{P}_x = \mathcal{P}_y = \text{maxf}$) and average optic flow pooling in the frontal visual field for vertical control ($\mathcal{P}_z = \text{avgf}$), see the dashed purple bars on Fig. 3.6 and Fig. 3.7.

With $\mathcal{P}_x = \mathcal{P}_y = \text{maxf}$ and $\mathcal{P}_z = \text{avgf}$, the predicted lateral position is the same as when maxf pooling is used on all axes (purple and blue dots on Fig. 3.7a). The predicted flight speed is

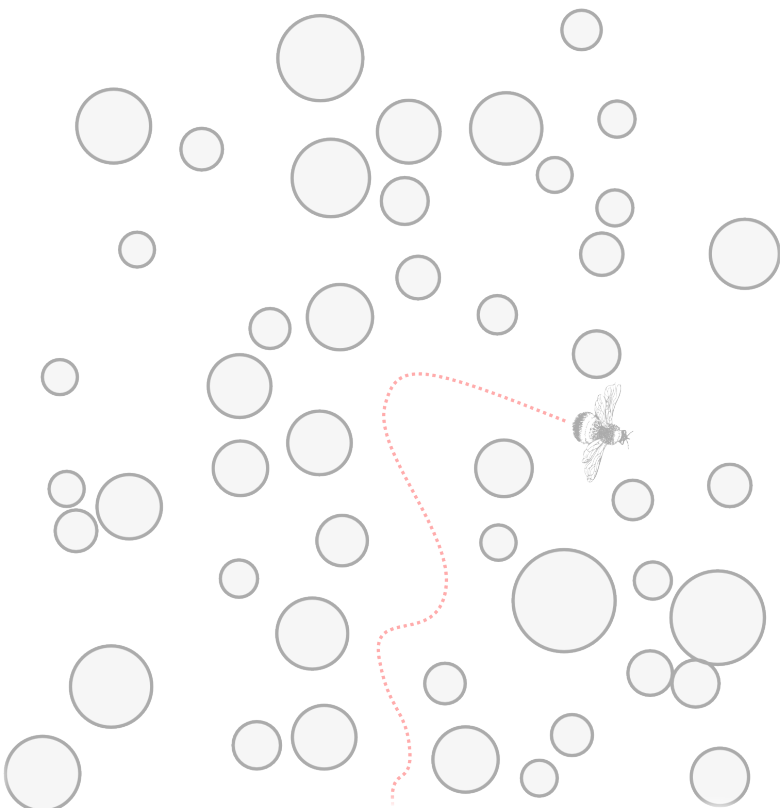
the same as when maxf pooling is used on all axes (purple and blue dots on Fig. 3.7c). The predicted vertical position is the same as when avgf pooling is used on all axes (purple and red dots on Fig. 3.7e). Furthermore, the contradiction described earlier is solved because the prediction made with $\mathcal{P}_x = \mathcal{P}_y = \text{maxf}$ and $\mathcal{P}_z = \text{avgf}$ achieves the lowest prediction error on all axes.

3.4 Conclusion

In this study, we investigate the effect of clutter on flight control in bumblebees and use mathematical models to understand how and where optic flow is being measured for this. We find that lateral position is controlled by balancing the maximum optic flow in the frontal visual field. This would ensure that even small obstacles in the flight path will be detected and used to control position, enabling rapid and effective collision avoidance responses. For speed control, the same pooling method is used to calculate the optic flow value that is then compared to a reference value, presumably set by the optimum sensitivity of specific speed-regulating neurons. Detecting the maximum optic flow output across an array of motion detectors is a biologically plausible operation which could easily be implemented in neural systems using a Winner-Take-All network [111] or using differentiation and zero-crossing [97], and is supported by previous behavioural [80] and analytical [Lecoeur2018] studies. Interestingly, we find that bees are most likely using a different pooling method for controlling vertical position. Instead of adjusting their height to maintain the ventral optic flow at a fixed reference value, as suggested for honeybees [99], we find that bumblebees regulate the ventral flow to a non-constant reference value equal to the average optic flow in the fronto-lateral visual field. It is possible that this finding is specifically related to our specific experimental situation which contained vertical obstacles that could not be avoided by changing vertical position. It would, therefore, be interesting for future investigations to examine the responses to horizontally-oriented obstacles and to model the pooling of lateral and ventral optic flow across varying azimuth and elevation angles. Taken together, our results suggest that bumblebees pool optic flow from the frontal visual field using two methods in parallel – averaging and maximum pooling – to control different aspects of flight. This has important implications for the design of flying robots because our findings suggest that a single forward pointed camera covering the frontal visual field is sufficient to replicate bumblebee behaviour.

4 Joint obstacle avoidance and visual navigation in cluttered environments

THIS CHAPTER presents a lightweight and reactive method for vision based navigation in cluttered environments designed by taking inspiration from previous findings on insect motion extraction and flight control in cluttered scenes. The method relies only on the local information extracted from panoramic images, and does not require prior knowledge about the position of the obstacles that the agent encounters on its path towards its navigation goal. By varying a single parameter that encodes the relative importance given to the angular size of gaps and their angular proximity to the navigation goal, a trade-off between flight safety and the directness of the generated trajectory can be found.



4.1 Introduction

Hymenopterans, such as bees, travel repeatedly during the course of the day between their nests and different food sites. They navigate in a wide variety of environments, ranging from relatively plain deserts to spatially cluttered forests, and on very different spatial scales, ranging from a few meters when pinpointing the nest entrance, to several kilometers on foraging flights [106].

During their trips, bees rely primarily on vision to control their flight and steer away from obstacles. Bees are known to extract the rate of apparent motion - optic flow - across the panoramic field of view of their low-resolution compound eyes. Several experiments demonstrated that optic flow is used to control lateral position, speed and height in corridors [41, 124, 1]. However, visually guided flights were never observed in controlled spatially cluttered environments.

In order to reach destination, bees steer toward their goal by combining egocentric cues, for example from an odometer, and allocentric information, for example from the visual panorama. If obstacles are on their paths, goal directed behavior needs to interact with collision avoidance. The interaction between goal and collision avoidance directions affects the flight strategy. Most existing flight control methods inspired by insects are designed to tackle collision avoidance but do not account for a navigation goal [102, 61, 63]. Goal oriented flight in cluttered environment is achieved by generating saccades which amplitude depend on the optic flow computed around the flying agent and on the goal direction [14]. The amplitude of the yaw saccades are computed so that the agent points in a direction that is the weighted sum of two direction: the goal direction, and the direction opposite to where obstacles are the closest on average. Because the proximity to obstacles is averaged all around the agent, however, there is no guaranty that this weighted sum will not result in the a saccade towards a small obstacle located in the direction opposite to that of average maximum proximity. Here, following the results of Chapter 3, we do not perform an averaging across the visual field, but instead rely on the location of the maximum rate of optic flow.

A lightweight and reactive method for vision based navigation in cluttered environments is presented. It takes inspiration from previous findings on insect motion extraction and flight control in cluttered scenes. The method relies only on the local information extracted from panoramic images, and does not require prior knowledge about the position of obstacles between the agent and its navigation goal. It is only assumed that the agent knows the direction to its navigation goal, which is achieved by insects through odometry [123] or polarized light sensing [65].

Chapter 4. Joint obstacle avoidance and visual navigation in cluttered environments

The method generates saccadic trajectories – i.e. succession of straight flight segments and fast yaw turns (the saccades) – similar to the trajectories performed by flying insects. During the straight flight segments, the agent does not perform body rotations, hence the perceived optic flow is inversely proportional to the distance to the surrounding obstacles. By locating the regions of its visual field where the optic flow amplitude is the highest, the agent locates the nearby obstacles. This is supported by a previous study where we showed that bumblebees selectively react to nearby obstacles by selecting the high rate of optic flow in the frontal visual field. The regions with lower optic flow amplitude between the detected obstacles form a set of gaps, whose centers are potential heading setpoints for the next saccade.

The amplitude of the saccades is chosen by rating each gap according to its apparent angular size and its angular proximity to the navigation goal. Because it relies on the maximum rate of optic flow, the method generates trajectories that are equidistant to the two closest obstacles, and are thus locally optimum in terms of safety. By varying a single parameter that encodes the relative importance given to the angular size of gaps and their angular proximity to the navigation goal, a trade-off between flight safety and the directness of the generated trajectory can be found.

4.2 Method

Let's consider a flying agent whose task is to navigate across a forest-like environment while avoiding collisions. We consider in this study that the agent flies on a two dimensional plane at constant altitude. The agent can control its flight in terms of its flight speed on the 2D plane V , and its heading angle Φ defined as the angle between north and the main axis of its body. At any time, the agent knows the goal heading Φ_{goal} that points to its destination. The way this information is obtained is outside the scope of this study, but it can be for example computed from visual odometry [123] or using polarized light cues [65].

We model a forest environment as a set of vertical "trees" with cylindrical shape. The trees are randomly distributed across the environment, and their position is unknown to the agent. We will refer to these trees as the obstacles that have to be avoided while navigating.

4.2.1 Safe paths in forest environments

In order to navigate safely through such environment, a flying agent should maximize the distance to obstacles. Considering n obstacles, the environment can be partitioned into n cells using Voronoi partitioning. Each cell of the resulting Voronoi diagram is defined as the set of points that are closer to one obstacle than to any other obstacle. The edges of the Voronoi cells

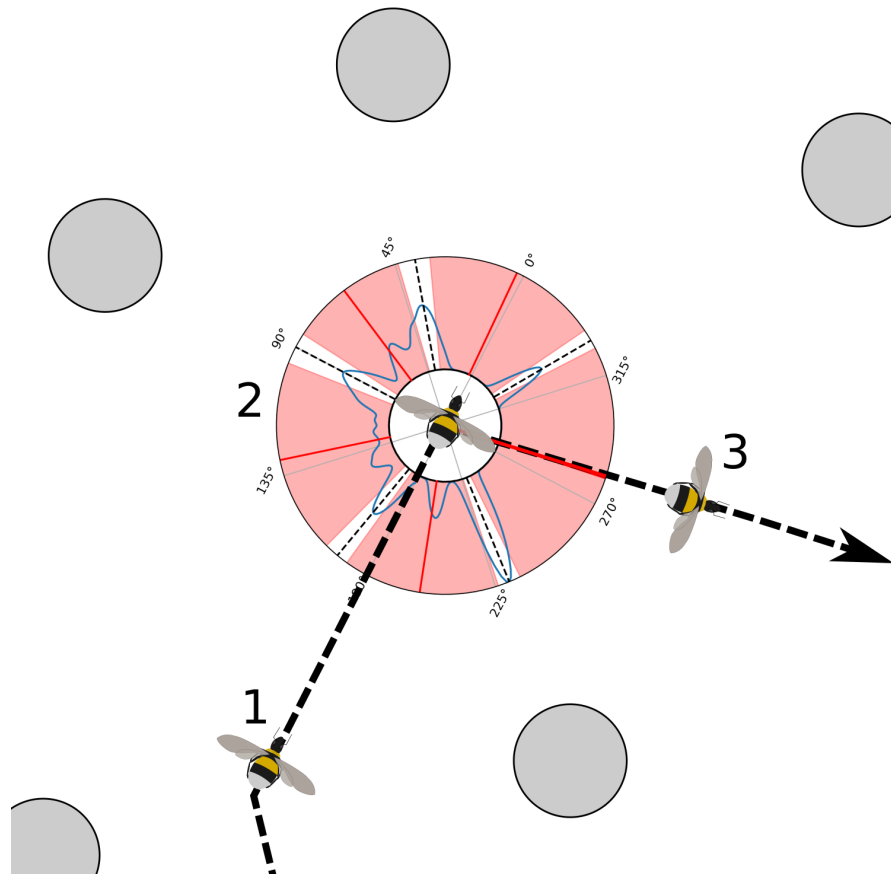


Figure 4.1 – Illustration of the navigation algorithm

The flying agent is here represented by a bumblebee. It flies along a saccadic trajectory made of straight segments and sharp yaw turns (dashed black line). Obstacles are represented by grey circles. (1): During a straight flight segment, the agent flies at constant speed and evaluates obstacle proximity around itself. (2): The field of obstacle proximity is displayed as the blue curve around the agent. Each pair of consecutive detected obstacles forms a gap (indicated by red regions). Each gap is defined by its center (red segments) and its angular width. Gaps are rated based on their relative widths, and on the angular distance between their centers and the desired navigation direction according to equation (4.5). (3): Among the five gaps present in this example, and thus the five potential headings, the agent has selected one and has performed a saccadic yaw turn. It then repeats the process and estimates obstacle proximity around itself.

Chapter 4. Joint obstacle avoidance and visual navigation in cluttered environments

of a forest environment are thus the points that are equidistant to the two closest obstacles. Consequently, Voronoi edges maximize the distance to the closest obstacles, and are locally the safest paths.

Voronoi partitioning is a classic tool for robot navigation in known environment. However our agent does not know the location of the obstacles and cannot build the voronoi diagram of the environment. Here we propose a navigation method that takes inspiration from Voronoi diagrams to follow safe paths, while not requiring a map of the environment.

4.2.2 Insect saccadic flight

Flying insects often exhibit a flight pattern called saccadic flight. A saccadic flight is a succession of straight segments and sharp turns called saccades. During straight flight segments, the heading and flight speed are held constant, with aligned heading and velocity vector. During a saccade, the insect performs a quick rotation on the yaw axis in order to modify its heading. It also generates lateral thrust in order to align its velocity vector to the next heading.

It is striking to note that an agent flying along a succession of Voronoi edges would perform saccadic flight. The edges of a Voronoi diagram are equivalent to the straight flight segments of an insect flight. The transition between two consecutive Voronoi edges is equivalent to a saccadic yaw turn. However it is unlikely that insects compute Voronoi diagrams to navigate. Most models of insect flight control strategies are simple reactive models based on optic flow.

Saccadic flight is beneficial for flight control strategies that rely on optic flow. Indeed optic flow can be decomposed into two components: the translational optic flow and the rotational optic flow [53, 64]. Translational optic flow is proportional to the ratio V/d between flight speed V and distance d to an obstacle. Thus it provides an estimate of the relative proximity of the environment. On the contrary, rotational optic flow is only function of the rotation rate of the observer, and does not contain information about the structure of the environment. Saccadic flight minimizes the time spent performing rotations, when optic flow cannot be used for navigation. Saccadic flight however maximizes the time spent in purely translational flight. It is believed that insects gather information about their surrounding environment during the straight flight segment, and modulate the amplitude of the saccade turn based on this information.

4.2.3 Obstacle detection

For a given flight speed and distance to an obstacle, the amplitude of optic flow depends on the location of the obstacle in the visual field.

$$\|\vec{O}F_{\text{trans}}(\alpha)\| = \frac{V}{d} \sin(\alpha) \quad (4.1)$$

where V is the flight speed, d is the distance to the obstacle, and α is the angle between the current direction of motion and a viewing direction. The translational optic flow is maximum for $\alpha = \pi/2$ and $\alpha = -\pi/2$, but is null for $\alpha = 0$ and $\alpha = \pi$. In other words, these regions of the visual field do not contain information on the proximity of obstacles and thus cannot be used. The relative proximity of obstacles is estimated by correcting the measurement of translational optic flow:

$$p(\alpha) = \frac{1}{V} \frac{\|\vec{O}F_{\text{trans}}(\alpha)\|}{\max(\epsilon, |\sin(\alpha)|)} \quad (4.2)$$

An obstacle is detected in the viewing direction α if the proximity in this direction is higher than a threshold set relative to the fastest moving object in the field of view:

$$\alpha \in \mathcal{O} \quad \text{if} \quad p(\alpha) \geq p_{\text{thres}}^{\text{rel}} \max_{[0,2\pi]}(p) \quad (4.3)$$

Conversely, gaps are defined as the regions between obstacles:

$$\alpha \in \mathcal{G} \quad \text{if} \quad p(\alpha) < p_{\text{thres}}^{\text{rel}} \max_{[0,2\pi]}(p) \quad (4.4)$$

where \mathcal{O} is the set of viewing directions where an obstacle is detected, \mathcal{G} is the set of viewing directions where obstacles are not detected, and $p_{\text{thres}}^{\text{rel}}$ is a relative proximity threshold. In our experiments, we use the value $p_{\text{thres}}^{\text{rel}} = 1/2$, which corresponds to selecting all obstacles that are at most twice as far as the closest obstacle. We define individual gaps \mathcal{G}_i as the connected regions of \mathcal{G} (Fig. 4.1). Each gap is defined by its center $\alpha_i^{\mathcal{G}}$ and its angular width $w_i^{\mathcal{G}}$, both expressed in radians.

4.2.4 Gap selection and saccade generation

We propose an approach where saccades are triggered periodically. During straight flight segment lasting Δt_{sac} seconds, optic flow is measured across the field of view, obstacle proximity is computed, obstacles and gaps are detected. At the end of the straight flight segment, the $N^{\mathcal{G}}$ detected gaps with centers $\alpha_i^{\mathcal{G}}$ and width $w_i^{\mathcal{G}}$ are evaluated and given a rating $r_i^{\mathcal{G}}$ according to

equation (4.5).

$$r_i^{\mathcal{G}} = \left(k_{\text{goal}} \left| (\Phi + \alpha_i^{\mathcal{G}}) - \Phi_{\text{goal}} \right| \right) + \left(k_{\text{width}} \left| \frac{w_i^{\mathcal{G}}}{\pi} \right| \right) \quad (4.5)$$

where k_{goal} and k_{width} are two gains that determine the relative importance of the direction of navigation and the width of the gap.

The gap with the highest rating is selected and the next heading setpoint is

$$\Phi_{k+1} = \Phi_k + \alpha_{\text{sel}}^{\mathcal{G}} \quad (4.6)$$

where Φ_k and Φ_{k+1} are the previous and next heading setpoints respectively; and $\alpha_{\text{sel}}^{\mathcal{G}}$ defines the center of the gap that received the highest rating.

4.3 Simulation Results

A simulated agent was flown in the same simulation environment as in Chapter 2, with the addition of vertical obstacles that are randomly located in a $6\text{m} \times 6\text{m}$ arena. The goal direction Φ_{goal} is set so as to let the agent navigate between four waypoints (WP1, WP2, WP3 and WP4 on Fig. 4.2a). Once the agent has reached a waypoint – which is defined by an acceptance radius of 1 meter – the goal direction is updated to point to the next waypoint. The trajectories of the simulated agent are recorded as it passes several times through the four waypoints.

The agent loosely follows the edges of the Voronoi graph of the environment, even though it does not have knowledge of the position of the obstacles and does not construct the Voronoi graph (Fig. 4.2a). As the gain k_{width} is increased, the mission time, defined as the time required to pass through the four waypoints, increases (Fig. 4.2b). This means that this gain can be used to affect the directness of the flight path. With larger values of k_{width} , and thus more emphasis on gap width, the trajectories will favor wide gaps, at the expense of mission time. An example can be seen with the purple curve on Fig. 4.2a, at location $(x = 0, y = 4)$, where the agent chose a longer path than with lower values of k_{width} in order to navigate through wider gaps. As the gain k_{width} is increased, the minimum distance to obstacles during the mission increases (Fig. 4.2c). This means that this gain can be used to select the level of flight safety. When $k_{\text{width}} = 0$, i.e. when gap width is not taken into account, the minimum distance to obstacles is very small, and collisions are more likely to occur. See for example the blue curve on Fig. 4.2 for a near miss at location $(x = 1, y = 0)$.

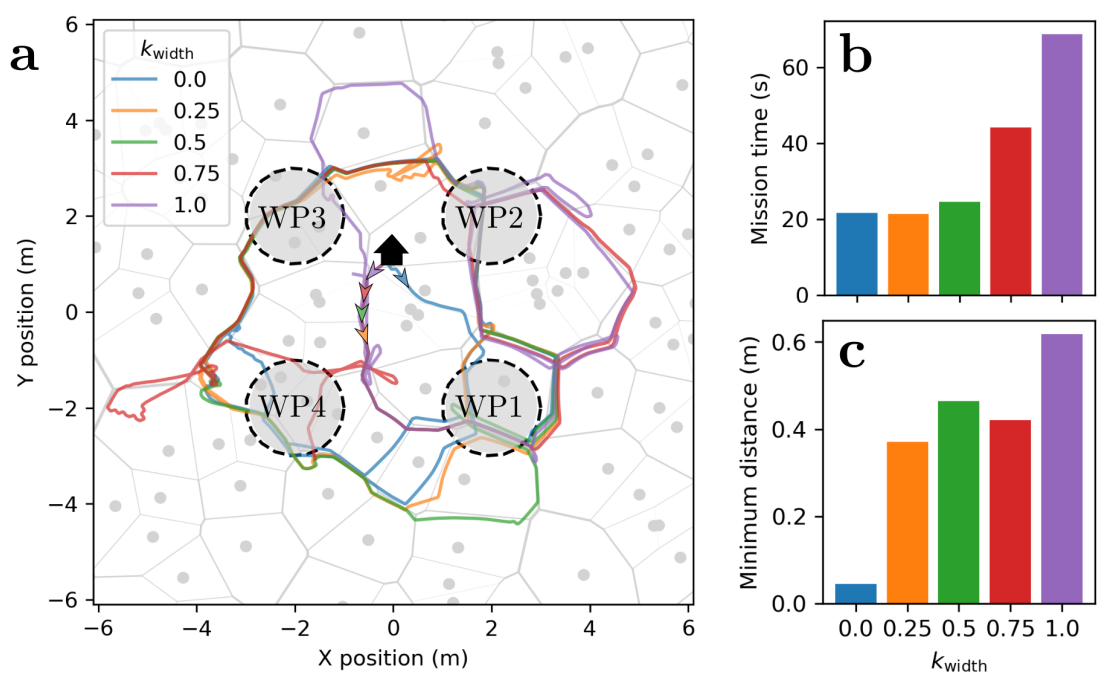


Figure 4.2 – Navigation results in simulation

(a): Top view of the forest-like environment. The obstacles are represented by grey circles. The edges of the Voronoi cells are displayed as grey segments with a thickness proportional to the distance between the edge and the two closest obstacles. In this example, all agents are initialized at location $(x = 0, y = 1)$ and heading $\Phi = 180^\circ$. They navigate towards the first waypoint (WP1) which is located at location $(x = 2, y = -2)$, until they reach an acceptance radius of 1 m around the waypoint, when they navigate towards the second waypoint (WP2), then the third waypoint (WP3), until they reach the fourth waypoint (WP4) and continue towards the first waypoint again. The gain k_{goal} is kept constant at a value of 1, the gain k_{width} is varied between 0 and 1. (b): Time required to reach the fourth waypoint (WP4) as a function of the weight k_{width} . The mission time increases with k_{width} , which denotes less direct flight paths. (c): Minimum distance between the agent and obstacles during the mission as a function of the weight k_{width} . The minimum distance increases with k_{width} , which denotes safer flight paths.

4.4 Hardware implementation

4.4.1 Design of a multirotor robot for saccadic flight

As stated before, saccadic flight allows to separate the phases of pure translational flight from the phases of fast heading change, called saccades. During translational flight phases, information about the proximity of the surrounding obstacles is easily obtained from translational optic flow (equation 4.2). During saccades, however, fast body rotations may blur images and the measured rotational optic flow does not depend on the proximity to the obstacles.

Most multirotor robots are under-actuated. They are equipped with a set of rotors laid out on a plane which, when actuated together, are capable of producing moments around the three body axes (roll, pitch and yaw), as well as a thrust force along the z body axis. However, most multirotor robots cannot produce thrust forces along their x and y body axes. As a consequence, in order to accelerate forward or sideways, they are required perform body rotations to align their z body axis with the desired thrust vector.

With such robot, a saccade requires three steps. First, the robot rotates in order produce a combination of vertical and lateral forces. Second, it accelerates sideways until the desired flight direction is achieved. Third, the robot rotates back to its original tilt angle in order to stop the saccade. While the second phase may be performed in a purely translational fashion, the tilt of the robot body has to be compensated for by the vision system (either by software or by a gimbal system) and it is not obvious that the field of view along the horizontal plane remains unobstructed during the manoeuver. Furthermore, the fact that the robot has to perform two body rotations for a single saccade is suboptimal because rotation phases do not allow to measure the proximity of obstacles.

In order to limit the amount of body rotations during saccadic flight and thus maximize the time spent in pure translation, we propose to design a fully actuated multirotor robot that is capable of producing thrust forces along its x and y body axes without tilting its body. With a fully actuated robot, a saccade may be composed of two phases only. First, the robot rotates about its yaw axis in order to align its x body axis with the desired velocity vector. Second, the robot generates lateral forces until its velocity vector matches the desired velocity vector.

We designed a robot with eight tilted rotors (Fig. 4.3). While six rotors would have been sufficient to control the six degrees of freedom of the robot, the two additional rotors make it more robust against actuator saturation and motor failure. The rotors are arranged on two layers. The rotors on the top layer are tilted inwards, while the rotors on the bottom layer are tilted outwards. This layout provides the fish eye camera that is mounted in the center with an

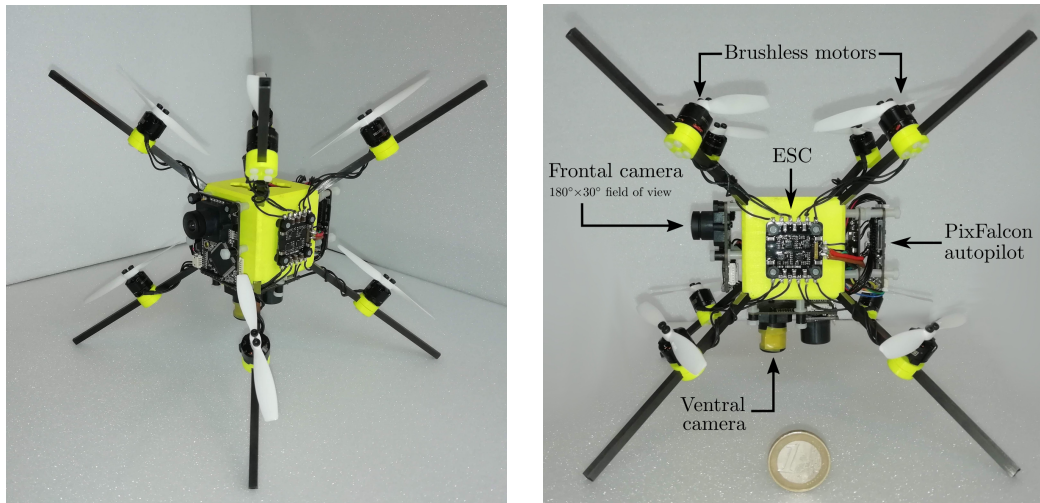


Figure 4.3 – BumbleBot: A lightweight flying robot for optic flow based navigation. The robot is equipped with 8 brushless motors tilted at 35°, either towards the center of the robot or away from it, which allows the generate horizontal thrust, and thus full controllability of the 6 degrees of freedom (roll, pitch, yaw, X, Y, Z). The robot is equipped with a downward facing optic flow camera (PX4Flow), associated to an ultrasonic range sensor, which provides a velocity estimate to the robot. In the front, a modified PX4Flow camera is mounted with a fish-eye lens, for a wide field of view of $180^\circ \times 30^\circ$. The total weight of the platform, battery included, is 169 grams.

unobstructed panoramic field of view at all time. Furthermore, the eight arms that connect the motors to the central body protect the propellers, cameras and other pieces of electronics in case of contact with obstacles.

4.4.2 Velocity control with level attitude

A schematic view of the implemented controller is presented on Figure 4.4. The optic flow computation is performed onboard the front-facing PX4Flow camera as described in previous chapters. The detection of obstacles is also performed onboard the front-facing camera, and the angular position of the obstacles is transmitted to the main autopilot board via a serial connection.

The saccade generator, velocity controller, attitude controller, control allocation, and estimator blocks are all running on the main PX4 flight controller (Fig. 4.4). The attitude controller, rate controller and estimator blocks are reusing the existing PX4 multirotor attitude controller and extended Kalman filter which are largely untouched. The saccade generator is implemented as a custom PX4 module as described in the methods section. The existing multirotor

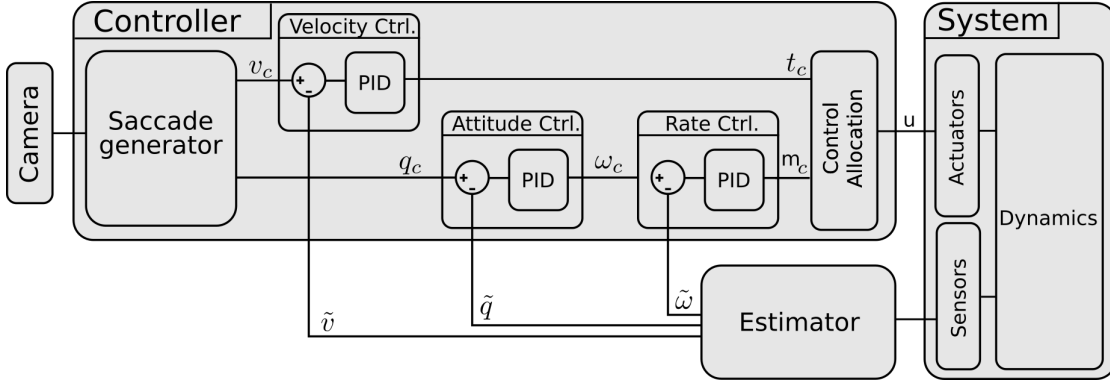


Figure 4.4 – Schematic view of the controller. The saccade generator receives the location of the surrounding obstacles from the camera. It generates a heading setpoint according to equation 4.5 which is converted into a velocity vector setpoint v_c and an attitude quaternion setpoint q_c . The 3D thrust setpoint t_c is computed by a PID controllers that regulates the robot's estimated velocity \tilde{v} to the velocity setpoint v_c . The 3D moment vector m_c is computed by two cascaded PID controllers that perform attitude and rate control. The 3D thrust setpoint and 3D moment setpoint are converted into actuator commands u in the control allocation step.

velocity controller is customized to provide level attitude setpoints and 3D thrust setpoints instead of tilted attitude setpoints and 1D thrust setpoints. The existing control allocation was customized to handle the control of six degrees of freedom.

At each saccade, the saccade generator increments the heading setpoint Φ_c . An attitude quaternion setpoint q_c with zero tilt angle (null roll and pitch angles) and a velocity setpoint \tilde{v}_c are computed as:

$$q_c = \begin{pmatrix} \cos(\Phi_c/2) \\ 0 \\ 0 \\ \sin(\Phi_c/2) \end{pmatrix} \quad \text{and} \quad \tilde{v}_c = \begin{bmatrix} v_{\text{cruise}} \left(1 - \frac{|\Phi_c - \tilde{\Phi}|}{\pi}\right) \cos(\Phi_c) \\ v_{\text{cruise}} \left(1 - \frac{|\Phi_c - \tilde{\Phi}|}{\pi}\right) \sin(\Phi_c) \\ k_z(p_{c_z} - \tilde{p}_z) \end{bmatrix} \quad (4.7)$$

where q_c is the attitude quaternion setpoint, Φ_c is the heading setpoint computed from the detected location of obstacles, $\tilde{\Phi}$ is the current heading, v_{cruise} is a reference cruise speed, which is scaled according to the heading error by a factor $\left(1 - \frac{|\Phi_c - \tilde{\Phi}|}{\pi}\right)$ in order to slow down before performing sharp turns, k_z is a proportional gain used to control the altitude, p_{c_z} is the altitude setpoint, and \tilde{p}_z is the current altitude.

The 3D thrust setpoint \tilde{t}_c is the output of a velocity PID controller. The 3D moment setpoint \tilde{m}_c is the output of two cascaded attitude and rate PID controllers. The thrust and moment

setpoints are combined into motor commands u in the control allocation block (Fig. 4.4).

4.4.3 Six degrees of freedom control allocation

We formulate the control allocation as a linear function from the six dimensional command space (3D moment and 3D force setpoints) to the n -dimensional actuator command space (one per rotor) represented by the control allocation matrix B .

$$\vec{u} = B \cdot \begin{bmatrix} \vec{m} \\ \vec{t} \end{bmatrix} \quad (4.8)$$

where \vec{t} is the (3×1) thrust setpoint vector, \vec{m} is the (3×1) moment setpoint vector, and \vec{u} is the $(n \times 1)$ actuator command vector.

The thrust and moment generated by the i -th rotor can be computed as

$$\begin{aligned} \vec{t}_i &= k_t \rho D^4 \omega_i^2 \vec{v}_i \\ \vec{m}_i &= (\vec{p}_i - \vec{p}_{cg}) \times (k_t \rho D^4 \omega_i^2 \vec{v}_i) - d_i k_m \rho D^5 \omega_i^2 \vec{v}_i \end{aligned} \quad (4.9)$$

where \vec{t}_i is the thrust generated by the i -th rotor, \vec{m}_i is the moment generated by the i -th rotor, \vec{v}_i is the main axis of the rotor (unit vector), \vec{p}_i is the position of the center of the rotor, \vec{p}_{cg} is the position of the center of mass of the robot, k_t is the thrust coefficient of the propeller, k_m is the moment coefficient of the propeller (usually $k_m \approx \frac{k_t}{10}$), ω_i is the rotation speed, ρ is the fluid density, D is the diameter of the propeller, and d_i is the rotation direction (-1 for clockwise rotation or +1 for counter clockwise rotation).

The thrust and moment generated by the i -th rotor can be approximated as

$$\begin{aligned} \vec{t}_i &\approx C_t u_i \vec{v}_i \\ \vec{m}_i &\approx (C_t (\vec{p}_i - \vec{p}_{cg}) \times \vec{v}_i) u_i - (d_i C_m \vec{v}_i) u_i \end{aligned} \quad (4.10)$$

where C_t is the approximated thrust coefficient of the motor-propeller unit, C_m is the approximated moment coefficient of the motor-propeller unit, and u_i is the motor command.

We define the actuator effectiveness matrix A as the $(6 \times n)$ matrix that allows to compute the thrust and moment generated by a set of n rotors as a function of the actuator command of each rotor.

$$\begin{bmatrix} \vec{m} \\ \vec{t} \end{bmatrix} = A \cdot \vec{u} \quad (4.11)$$

The matrix A can be found by concatenating the moment and thrust vectors generated by the n rotors:

$$\begin{bmatrix} m_x \\ m_y \\ m_z \\ t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} m_x^0 & \dots & m_x^i & \dots & m_x^{n-1} \\ m_y^0 & \dots & m_y^i & \dots & m_y^{n-1} \\ m_z^0 & \dots & m_z^i & \dots & m_z^{n-1} \\ t_x^0 & \dots & t_x^i & \dots & t_x^{n-1} \\ t_y^0 & \dots & t_y^i & \dots & t_y^{n-1} \\ t_z^0 & \dots & t_z^i & \dots & t_z^{n-1} \end{bmatrix} \cdot \begin{bmatrix} u_0 \\ \vdots \\ u_i \\ \vdots \\ u_{n-1} \end{bmatrix} \quad (4.12)$$

Finally, the control allocation matrix B is computed as the Moore-Penrose pseudo-inverse of matrix A :

$$B = A^+ \quad (4.13)$$

The singular value decomposition (SVD) of A gives $A = U \cdot \sigma \cdot V^T$, where σ is a diagonal matrix. If A has a rank r , then the first r elements of σ are non-nul. B can be computed as $B = V \cdot \sigma^+ \cdot U^T$, where σ^+ is a diagonal matrix that contains the inverse of the non-nul terms of the diagonal of σ .

For robots where the position and orientation of the rotors are known, and where the thrust and moment constants can be measured, B can be pre-computed offline. An automatic generator was implemented and is now used by default in the PX4 autopilot [90]. The generator takes as input a text file that describes the geometry of the robot, i.e. the list of rotors with their respective location, orientation and constants. Then, the generator pre-computes the control allocation matrix and generates C++ code to be used by the PX4 multirotor mixer.

4.4.4 Results and limitations

The navigation strategy was tested on the flying robot in an experimental arena of dimension 11m \times 12m containing 8 obstacles of dimension 0.35m \times 0.35m \times 1.6m (Fig. 4.5). The robot was flown manually to several starting locations, then was flown autonomously using the proposed navigation algorithm. Each trial was stopped manually either when the robot flew towards the limits of the experimental arena, or when a collision with an obstacle occurred.

For this experiment, goal-oriented navigation was disabled ($k_{\text{goal}} = 0$) due to the relatively small dimension of the experimental room. Thus, the robot only relies on the gap width ($k_{\text{width}} = 1$) to navigate. The robot was flown at a cruise speed $v_{\text{cruise}} = 0.5\text{m/s}$, an altitude setpoint $p_{cz} = 0.6\text{m}$, and a minimum inter-saccade time $\Delta t_{\text{sac}} = 1\text{s}$.

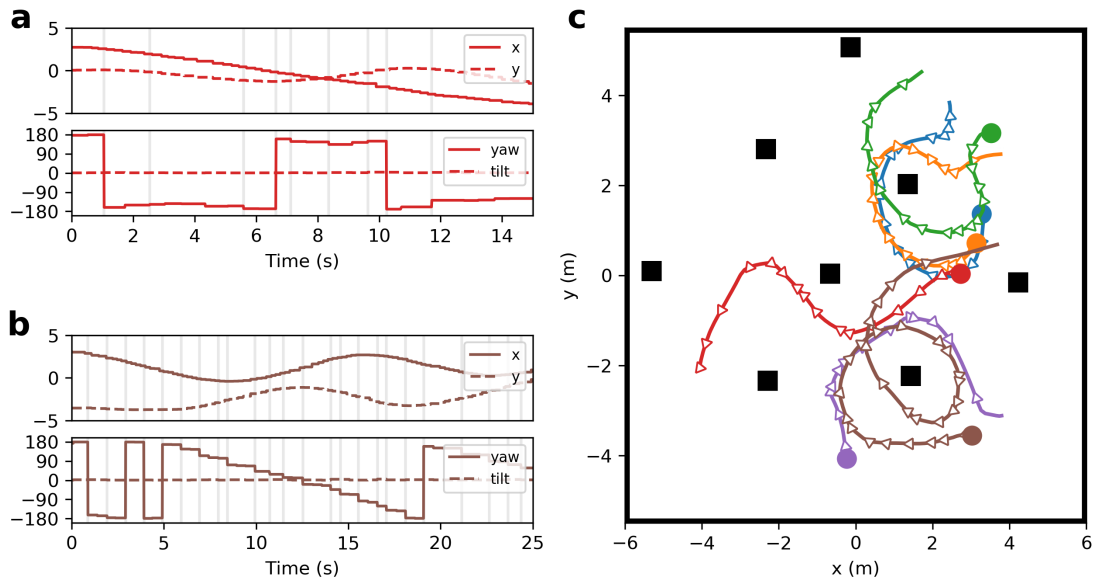


Figure 4.5 – Sample trajectories recorded during hardware experiments
(a and b): State of the robot against time during two autonomous flights. The horizontal position of the robot (top graphs) varies as the robots navigates around obstacles. Note that the robot attitude remains level (bottom graphs, dashed line), while the yaw angle is incremented by steps at each saccade (bottom graphs, solid line). **(c):** Top view of the flight arena with the obstacles depicted as black squares and trajectories shown as colored curves. The colored dots indicate the starting position of the robot and the white triangles represent the position and orientation of the robot each time a saccade was performed.

Chapter 4. Joint obstacle avoidance and visual navigation in cluttered environments

It appears that –similar to the simulated paths– the robot followed several times the same paths during different trials. For example, the blue and orange tracks are almost entirely superimposed, and the purple track follows a portion of the the brown track before it diverges (Fig. 4.5c).

It was also noted that when an experiment was started with the robot facing an obstacle, the robot would collide frontally into the obstacle most of the times. On the contrary, when an experiment was started with the robot facing a gap between two obstacles, the flights were much longer, usually ending with the robot flying towards one of the white walls in the experimental room. We believe that frontal collisions are due to the fact that the frontal region is effectively a blind spot for optic flow based methods. Indeed, as can be seen in equation (4.2), as the robot always aligns its heading with the velocity setpoint, the angle α between viewing direction and flight direction is very small in the frontal region, and it is thus impossible to obtain a proximity estimate. A potential solution to alleviate this limitation is to introduce lateral oscillations during the inter-saccadic flight phases. This would allow to measure non zero optic flow in the frontal visual field and hence remove the blind spot and allow the robot to avoid frontal collisions.

4.5 Conclusion

In this chapter, we tackled insect-inspired navigation in cluttered environments with a novel navigation algorithm based on the detection of the fastest moving objects in the visual field. Our method uses the location of the fastest moving objects around a flying agent in order to generate yaw saccades towards a flight direction that is free of obstacles and that leads to the desired navigation goal. The algorithm was validated with a simulated agent navigating in a cluttered environment. Future work will involve further validation on real hardware (Fig. 4.3). Initial hardware experiments were performed and showed that the method can be applied to real world indoor scenarios with well-contrasted obstacles. Further tests on the flying robot will include the testing and characterization of the obstacle detection and the effect of algorithm parameters in natural environments such as forests. It will also be interesting to study how lateral oscillation without body rotation –which are feasible with the design presented on Fig. 4.3– can improve the estimation of proximity in the forward direction, which is now a blind area because of the lack of optic flow. By adding lateral oscillations, the focus of optic flow expansion is shifted laterally and a proximity estimate can be obtained in the forward direction.

5 Concluding remarks

THIS chapter summarizes the main contributions of this thesis in the topic of insect inspired visual perception, flight control, and collision avoidance. Novel algorithms for optic flow estimation, visual flight control and navigation were proposed and validated with behavioural studies, computer simulations, and robotic experiments. The main accomplishments of this thesis are summarized in Section 5.1. Potential direction for future work are given in Section 5.2.



5.1 Main accomplishments

The goal of this thesis was to study the behaviour of flying insects in cluttered environment in order to enhance our knowledge on insect visual perception and control, and to allow the design of efficient and lightweight flying robots. This section summarizes the three main contributions that can be extracted from this work. The first contribution was the research done on the Elementary Motion Detector model and the development of an optic flow estimation algorithm that builds on top of the EMD. The second contribution was the modelling of bumblebee flight control in cluttered environment, which highlighted the importance of optic flow pooling and showed that maximum pooling is likely used by flying insects to avoid obstacles. Finally, the last contribution was a novel algorithm that models how insects may navigate in cluttered environments by using optic flow to locate the surrounding obstacles and perform saccadic manoeuvres. Each contributed model and algorithm was validated in simulation or on autonomous flying robots.

Insect-inspired estimation of optic flow

The Elementary Motion Detector is a visual motion estimator that, although likely present in insect neural system, is poorly correlated to optic flow. We showed that the location of the maximum response in an EMD array provides appropriate estimation of optic flow when the agent is flying sufficiently fast and/or close to the surface. We proposed a flight control strategy that uses the location of maximum EMD response as control input instead of optic flow and tested it in a 3D simulation and on a flying robot. Our strategy is both a good candidate for the neuronal circuitry used in insect brain for visual flight control, and a lightweight alternative to existing algorithms for optic flow estimation.

The role of optic flow pooling in insect flight control

Insect visual flight control has been extensively tested in numerous experimental environments with varying sizes, shapes, and wall texture, but much less was known about insect visual flight control in cluttered environments. We investigated the effect of clutter on flight control in bumblebees and used mathematical models to understand how and where optic flow is being measured. Our results suggest that lateral position is controlled by balancing the maximum optic flow in the frontal visual field. In other words, unlike current models that consider static optic flow pooling, bumblebees appear to be selectively reacting to the fastest moving objects in their frontal visual field.

Joint obstacle avoidance and navigation in cluttered environments

Foraging insects like bees routinely traverse cluttered environments on their way between the hive and a food source. It is still unclear how they are able to navigate and avoid collision in unknown and cluttered environments. Building on the findings of this thesis in experimental corridor, we proposed a novel navigation algorithm based on the detection of the fastest moving objects in the visual field. The algorithm was validated with a simulated agent navigating in a cluttered environment.

5.2 Future work and outlook

5.2.1 Combining deep learning methods with bio-inspired models

Reinforcement learning is used in [108] to train a vision-based controller and to guide a quadcopter safely through a forest. Input from a human is used to iteratively refine a controller linking predefined image descriptors to lateral steering commands. The method shows promising results and performs well even in environments where it received no training. The major limitation of this method comes from the fact that all the visual processing is performed on a computer on the ground and would be difficult to embed on a smaller platform.

Recent years have seen tremendous progress in the field of computer vision with the increasing performance of deep convolutional neural networks [76]. Deep neural networks were successfully employed to learn vision based controllers. A dataset of images gathered by humans during hikes allowed to learn whether the track was located on the right or left side of the camera and thus let a multicopter follow a forest trail autonomously [54]. By repeatedly letting a drone crashing and by recording the last video frames before the collision, a deep neural network was trained to predict collisions, which enabled collision avoidance in complex indoor environments [51]. Finally, videos from cars were used to train a neural network to navigate in urban environments [87].

With the miniaturization of single board computers, deep learning has become accessible to small flying robots. However, deep learning require very large amount of training data, which is often not convenient to obtain with flying robots. On the contrary, it is relatively easy to obtain data from flying insects, and insects are systems that are the result of million years of evolutionary optimization. Surely one could explore how to take advantage from deep learning techniques to model insect flight.

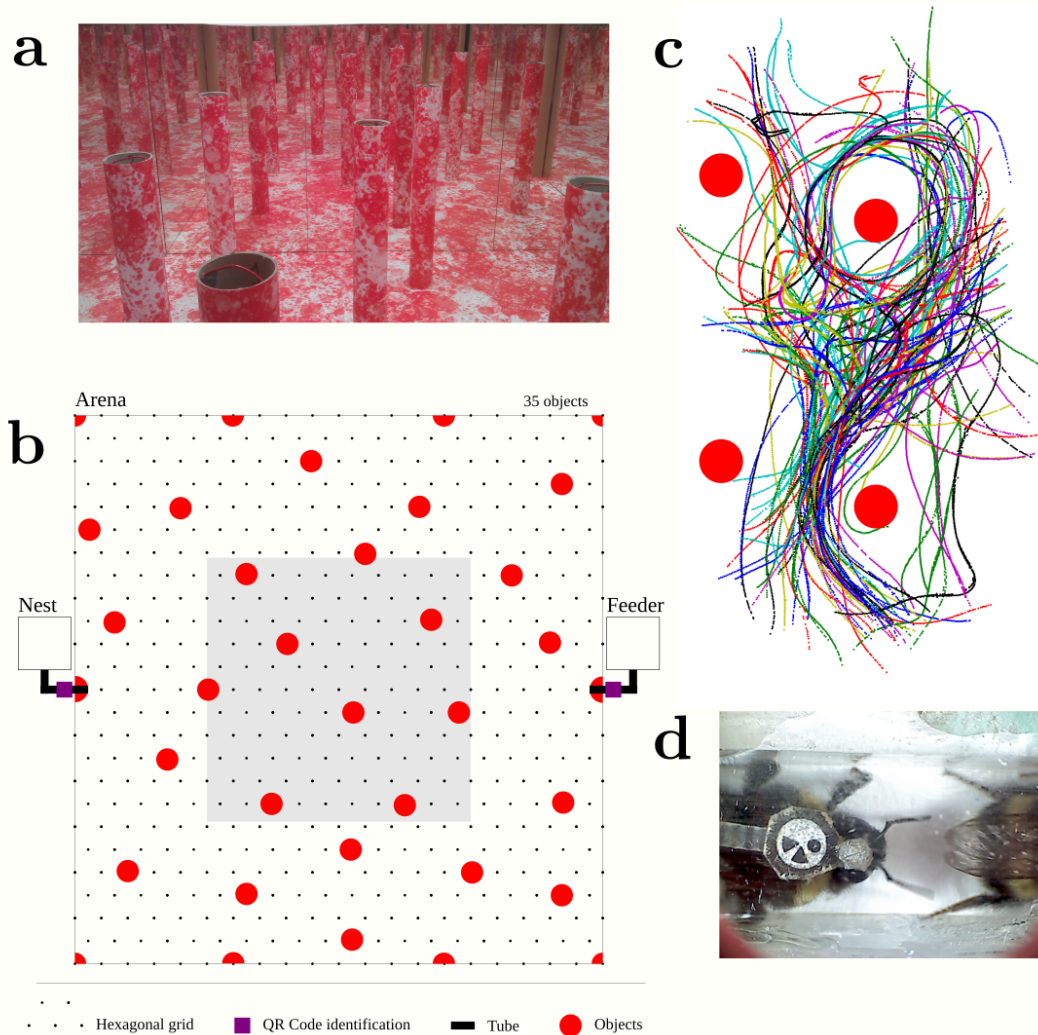


Figure 5.1 – Experimental setup for bumblebee navigation in cluttered environment (a): Picture of the experimental setup. Vertical cylinders are randomly located in the arena. The floor and cylinders are covered with a dead leaf pattern similar to the one used throughout this thesis. The walls are covered with mirrors, which give the illusion that the environment is very large, thus simulating an infinite forest and avoiding border effects. (b): Top view of the experimental setup. The bumblebee nest and the feeder are located at two opposite ends of the arena and are connected to the inside of the arena via tubes where the bumblebees can crawl. On this tube is mounted a camera system to identify the bumblebees as they enter and leave the arena. (c): Sample trajectories recorded using a motion capture system and reflective markers mounted on the back of the bumblebees. The trajectories presented here are the equivalent of half a day of experiment. (d): Picture taken by the automated bumblebee identification system at the tube leading to the nest. A visual tag is visible on the back the bumblebee, as well as an infrared reflective marker. The visual tag is used to identify the bumblebee, while the reflective marker is used for trajectory tracking using a motion capture system.

5.2.2 Behavioural study of insect goal-oriented flight in cluttered environments

This thesis provided the first behavioural study of bumblebees flight control in corridors containing obstacles (Chapter 3). The main finding of this study was that bumblebee use the maximum optic flow in the frontal visual field to control their flight laterally. Based on this finding, we extrapolated a method for optic flow based navigation in forest-like cluttered environments (Chapter 4). However, whether bees actually use such method in real life should be tested experimentally.

In order to do this, an experimental setup was designed in collaboration with Dr. Olivier Bertrand, Prof. Martin Egelhaaf, and the Lund Vision Group (Fig. 5.1a). The setup allowed to record the 3D trajectories of bumblebees traversing an artificial forest-like environments between their nest and a food source (Fig. 5.1c). The goal was to reconstruct the flow field perceived by the bumblebees during the flights and study how the visual motion information was converted into flight commands, in a similar fashion to Chapter 3.

A potential solution to process the experimental data might be machine learning techniques. These techniques usually require a large amount of data that can be obtained with an automated setup such as the one presented on Fig. 5.1b. For example, one may formulate the problem as a regression problem like in [113]. The goal would be to use the data gathered from real bumblebee flights to train a deep neural network to predict a heading command based on high dimensional optic flow input. This solution would however introduce new challenges in the interpretability of the learned network [136].

A Publications

Manuscripts published in peer-reviewed scientific journals and conferences:

- [72] LecoEUR, J., Baird, E. & Floreano, D. Spatial Encoding of Translational Optic Flow in Planar Scenes by Elementary Motion Detector Arrays. *Scientific Reports* **8**, 5821 (2018).
- [70] LecoEUR, J., Baird, E. & Floreano, D. *A Bee in the Mirror: A Bio-Inspired Model for Vision Based Mid-Air Collision Avoidance (poster)* in *RIN13 Bionav - The application of animal navigation techniques in autonomous vehicles*, Royal Holloway College (2013).
- [96] Pericet-Camara, R., Bahi-Vila, G., LecoEUR, J. & Floreano, D. *Miniature artificial compound eyes for optic-flow-based robotic navigation* in *2014 13th Workshop on Information Optics, WIO 2014* (2014).
- [30] Daler, L., LecoEUR, J., Hahlen, P. B. & Floreano, D. A flying robot with adaptive morphology for multi-modal locomotion. *IEEE International Conference on Intelligent Robots and Systems*, 1361–1366 (2013).

Manuscripts under review in peer-reviewed scientific journals:

- [74] LecoEUR, J., Dacke, M., Floreano, D. & Baird, E. The role of optic flow pooling in insect flight control in cluttered environments. *Scientific Reports* **in review** (2018).

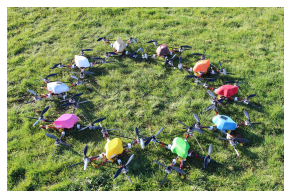
Manuscripts in preparation for future publication:

- [73] LecoEUR, J., Baird, E. & Floreano, D. Visual acuity requirements for mid-air head-on collision avoidance: a scaling law study. (*in preparation*).

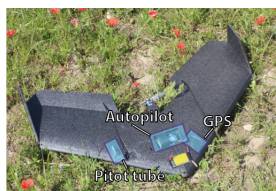
Appendix A. Publications

- [71] Lecoer, J., Baird, E. & Floreano, D. Insect inspired algorithm for joint obstacle avoidance and navigation in forest-like environment. (*in preparation*).
- [113] Schilling, F., Lecoer, J., Schiano, F. & Floreano, D. Learning Vision-based Cohesive Flight in Drone Swarms. arXiv: 1809.00543 (2018).
- [75] Lecoer, J., Dousse, N., Heitz, G., Schill, F., L'Éplattenier, G. & Huber, B. MAV'RIC: Open source software library to build drone autopilots. (*in preparation*).

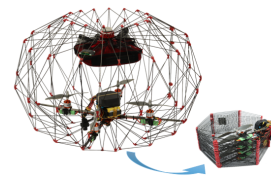
B MAV'RIC: Open source software library for the development of drones



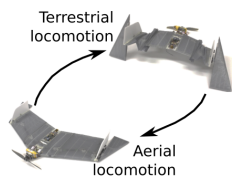
(a) LEQuad: a swarm of 10 quadrotors using the MAV'RIC framework in an outdoor environment.



(b) LEWing: a fixed wing using the MAV'RIC framework.



(c) PackDrone: a foldable drone inspired by origami. The volume of the folded robot is reduced by 92%.



(d) DALER: a multi-modal robot capable of flying, hovering and crawling using a minimal set of actuators.



(e) LEQuad equipped with two fisheye cameras, it avoids collisions using optic flow computed across its 360 degrees field of view



(f) YWing: a tailsitter VTOL capable of insect-like saccadic flight.

Figure B.1 – Research platforms built using the MAV'RIC software library.

MAV'RIC is an open-source software library to build autopilots for small drones that was developed during this thesis [75]. MAV'RIC framework is designed with a focus on modularity, flexibility and explicitness. It is aimed for researchers requiring a large freedom to meet the specific requirements of their research projects. Even if MAV'RIC is primarily a software library, it provides ready-to-use sample projects. The library code is shared between research projects, and project code is specific to a particular application. As a consequence, the researcher can write his or her own application by importing parts of the library in the project code

Appendix B. MAV'RIC: Open source software library for the development of drones

without interfering with other researchers. In addition, he or she can develop new features and incorporate them into the library such that the whole group benefits from the development.

MAV'RIC [75] was used in several research projects in the Laboratory of Intelligent Systems and contributed to the following publications:

- [30] Daler, L., Lecoeur, J., Hahlen, P. B. & Floreano, D. A flying robot with adaptive morphology for multi-modal locomotion. *IEEE International Conference on Intelligent Robots and Systems*, 1361–1366 (2013).
- [35] Dousse, N., Heitz, G. & Floreano, D. Extension of a ground control interface for swarms of Small Drones. *Artificial Life And Robotics* **21**, 9. 308–316 (2016).
- [34] Dousse, N., Heitz, G. H. M., Schill, F. S. & Floreano, D. Human-Comfortable Collision Free Navigation for Personal Aerial Vehicles. *IEEE Robotics and Automation Letter* **2**, 8. 358–365 (2017).
- [32] di Luca, M., Mintchev, S., Heitz, G. H. M., Noca, F. & Floreano, D. Bioinspired morphing wings for extended flight envelope and roll control of small drones. *Interface Focus* **7**, 11. 20160092 (2017).
- [91] Miehlbradt, J. C., Cherpillod, A. T., Mintchev, S., Coscia, M., Artoni, F., Floreano, D., *et al.* Data-driven body-machine interface for the accurate control of drones. *Proceedings of the National Academy of Sciences*, 201718648 (2018).
- [107] Rognon, C., Mintchev, S., Dell'Agnola, F. I. T., Cherpillod, A. T., Atienza Alonso, D. & Floreano, D. FlyJacket: An Upper Body Soft Exoskeleton for Immersive Drone Control. *IEEE Robotics and Automation Letters* **3**, 2362–2369 (2018).
- [12] Basiri, M., Schill, F. S., Lima, P. & Floreano, D. On-Board Relative Bearing Estimation for Teams of Drones Using Sound. *IEEE Robotics and Automation Letters* **1**, 820–827 (2016).
- [13] Basiri, M., Schill, F., Lima, P. & Floreano, D. Localization of emergency acoustic sources by micro aerial vehicles. *Journal of Field Robotics* (2017).

The source code¹, documentation², user guide³ and associated hardware and electronic designs⁴ are available online.

¹https://github.com/lis-epfl/MAVRIC_Library

²http://lis-epfl.github.io/MAVRIC_Library/doxygen/classes.html

³http://lis-epfl.github.io/MAVRIC_Library/

⁴<https://github.com/lis-epfl/MAVRIC>

Bibliography

- [1] Altshuler, D. L. & Srinivasan, M. V. Comparison of Visually Guided Flight in Insects and Birds. *Frontiers in Neuroscience* **12** (2018).
- [2] Aptekar, J. W., Shoemaker, P. A. & Frye, M. A. Figure tracking by flies is supported by parallel visual streams. *Current Biology* **22**, 482–487 (Mar. 2012).
- [3] Bahl, A. Object tracking in motion-blind flies. *Nature Neurosci.* **16**, 1–11 (2013).
- [4] Baird, E. Visual control of flight speed in honeybees. *Journal of Experimental Biology* **208**, 3895–3905 (Oct. 2005).
- [5] Baird, E., Boeddeker, N., Ibbotson, M. R. & Srinivasan, M. V. A universal strategy for visually guided landing. *Proceedings of the National Academy of Sciences of the United States of America* **110**, 18686–18691 (2013).
- [6] Baird, E. & Dacke, M. Visual flight control in naturalistic and artificial environments. *Journal of Comparative Physiology A* **198**, 869–876 (2012).
- [7] Baird, E., Kornfeldt, T. & Dacke, M. Minimum viewing angle for visually guided ground speed control in bumblebees. *Journal of Experimental Biology* **213**, 1625–1632 (May 2010).
- [8] Balboa, R. M. & Grzywacz, N. M. Power spectra and distribution of contrasts of natural images from different habitats. *Vision Research* **43**, 2527–2537 (2003).
- [9] Barlow. in *Sensory Communication* June 2015, 844 (2012).
- [10] Barlow, H. B. & Levick, W. R. The mechanism of directionally selective units in rabbit's retina. *The Journal of Physiology* **178**, 477–504 (1965).
- [11] Barnett, P. D., Nordström, K. & O'Carroll, D. C. Motion adaptation and the velocity coding of natural scenes. *Current Biology* **20**, 994–999 (2010).
- [12] Basiri, M., Schill, F. S., Lima, P. & Floreano, D. On-Board Relative Bearing Estimation for Teams of Drones Using Sound. *IEEE Robotics and Automation Letters* **1**, 820–827 (2016).

Bibliography

- [13] Basiri, M., Schill, F., Lima, P. & Floreano, D. Localization of emergency acoustic sources by micro aerial vehicles. *Journal of Field Robotics* (2017).
- [14] Bertrand, O. J., Lindemann, J. P. & Egelhaaf, M. A Bio-inspired Collision Avoidance Model Based on Spatial Information Derived from Motion Detectors Leads to Common Routes. *PLoS Computational Biology* **11**, 1–28 (2015).
- [15] Beyeler, A., Zufferey, J. C. & Floreano, D. *Vision-based control of near-obstacle flight in Autonomous Robots* **27** (Aug. 2009), 201–219.
- [16] Bhagavatula, P. S., Claudianos, C., Ibbotson, M. R. & Srinivasan, M. V. Optic flow cues guide flight in birds. *Current Biology* **21**, 1794–1799 (Nov. 2011).
- [17] Borst, A. Models of motion detection. *Nature neuroscience* **3 Suppl**, 1168 (2000).
- [18] Borst, A. Fly visual course control: behaviour, algorithms and circuits. *Nature reviews. Neuroscience* **15**, 590–599 (2014).
- [19] Borst, A. In search of the holy grail of fly motion vision. *European Journal of Neuroscience* **40**, 3285–3293 (2014).
- [20] Borst, A. Neural Circuits for Elementary Motion Detection. *Journal of neurogenetics* **7063**, 1–13 (2014).
- [21] Braun, E., Dittmar, L., Boeddeker, N. & Egelhaaf, M. Prototypical components of honeybee homing flight behavior depend on the visual appearance of objects surrounding the goal. *Frontiers in behavioral neuroscience* **6**, 1 (Jan. 2012).
- [22] Braun, E., Geurten, B. & Egelhaaf, M. Identifying prototypical components in behaviour using clustering algorithms. *PLoS ONE* **5**, e9361 (Jan. 2010).
- [23] Brinkworth, R. S. A. & O’Carroll, D. C. Robust models for optic flow coding in natural scenes inspired by insect biology. *PLoS Computational Biology* **5** (2009).
- [24] Briod, A., Zufferey, J. C. & Floreano, D. A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments. *Autonomous Robots* **40**, 789–803 (2016).
- [25] Bry, A. & Roy, N. *Rapidly-exploring random belief trees for motion planning under uncertainty* in *Proceedings - IEEE International Conference on Robotics and Automation* (Ieee, May 2011), 723–730.
- [26] Buchner, E. in *Photoreception and Vision in Invertebrates* 561–621 (1984).
- [27] Cao, F., Guichard, F. & Hornung, H. *Dead leaves model for measuring texture quality on a digital camera* in *Proc. SPIE 7537, Digital Photography VI, 75370E* (eds Imai, F., Sampat, N. & Xiao, E) (Jan. 2010), 75370E.

- [28] Chakravarthi, A., Kelber, A., Baird, E. & Dacke, M. High contrast sensitivity for visually guided flight control in bumblebees. *Journal of Comparative Physiology A* (2017).
- [29] D'Andrea, R. Can Drones Deliver? *Article in IEEE Transactions on Automation Science & Engineering* **11**, 647–648 (2014).
- [30] Daler, L., Lecoq, J., Hahlen, P. B. & Floreano, D. A flying robot with adaptive morphology for multi-modal locomotion. *IEEE International Conference on Intelligent Robots and Systems*, 1361–1366 (2013).
- [31] David, C. T. Compensation for height in the control of groundspeed by *Drosophila* in a new, 'Barber Pole' wind tunnel. *Journal of Comparative Physiology A* **147**, 485–493 (1982).
- [32] Di Luca, M., Mintchev, S., Heitz, G. H. M., Noca, F. & Floreano, D. Bioinspired morphing wings for extended flight envelope and roll control of small drones. *Interface Focus* **7**, 11. 20160092 (2017).
- [33] Dickson, W. B., Straw, A. D., Poelma, C. & Dickinson, M. H. An Integrative Model of Insect Flight Control. *44th AIAA Aerospace Sciences Meeting and Exhibit; Reno, NV; USA; 9-12 Jan*, 1–19 (2006).
- [34] Dousse, N., Heitz, G. H. M., Schill, F. S. & Floreano, D. Human-Comfortable Collision Free Navigation for Personal Aerial Vehicles. *IEEE Robotics and Automation Letter* **2**, 8. 358–365 (2017).
- [35] Dousse, N., Heitz, G. & Floreano, D. Extension of a ground control interface for swarms of Small Drones. *Artificial Life And Robotics* **21**, 9. 308–316 (2016).
- [36] Dror, R. O., O'Carroll, D. C. & Laughlin, S. B. Accuracy of velocity estimation by Reichardt correlators. *Journal of the Optical Society of America A* **18**, 241 (2001).
- [37] Dyhr, J. P. & Higgins, C. M. The spatial frequency tuning of optic-flow-dependent behaviors in the bumblebee *Bombus impatiens*. *The Journal of experimental biology* **213**, 1643–1650 (2010).
- [38] Egelhaaf, M. & Reichardt, W. Dynamic response properties of movement detectors: Theoretical analysis and electrophysiological investigation in the visual system of the fly. *Biological Cybernetics* **56**, 69–87 (1987).
- [39] Egelhaaf, M. & Borst, A. Transient and steady-state response properties of movement detectors. *Journal of the Optical Society of America a-Optics Image Science and Vision* **6**, 116–127 (1989).
- [40] Egelhaaf, M., Boeddeker, N., Kern, R., Kurtz, R. & Lindemann, J. P. Spatial vision in insects is facilitated by shaping the dynamics of visual input through behavioral action. *Frontiers in Neural Circuits* **6**, 108 (Jan. 2012).

Bibliography

- [41] Egelhaaf, M. & Kern, R. Vision in flying insects. *Current Opinion in Neurobiology* **12**, 699–706 (Dec. 2002).
- [42] Eichner, H., Joesch, M., Schnell, B., Reiff, D. F. & Borst, A. Internal Structure of the Fly Elementary Motion Detector. *Neuron* **70**, 1155–1164 (2011).
- [43] Expert, F. & Ruffier, F. Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers. *Bioinspiration & Biomimetics* **10**, 26003 (2015).
- [44] Filliat, D. & Meyer, J.-a. Map-based navigation in mobile robots - II. A review of map-learning and path-planning strategies. *Journal of Cognitive Systems Research* **4**, 243–282 (2003).
- [45] Floreano, D., Ijspeert, A. J. & Schaal, S. Robotics and neuroscience. *Current Biology* **24**, R910–R920 (2014).
- [46] Floreano, D. & Wood, R. J. Science, technology and the future of small autonomous drones. *Nature* **521**, 460–466 (2015).
- [47] Floreano, D. & Zufferey, J. C. Insect vision: A few tricks to regulate flight altitude. *Current Biology* **20**, R847–R849 (2010).
- [48] Franceschini, N., Pichon, J. M., Blanes, C. & Brady, J. M. *From Insect Vision to Robot Vision [and Discussion]* 1992.
- [49] Franz, M. O. & Krapp, H. G. Wide-field, motion-sensitive neurons and matched filters for optic flow fields. *Biological Cybernetics* **83**, 185–197 (2000).
- [50] Gabbiani, F., Krapp, H. G., Hatsopoulos, N., Mo, C. H., Koch, C. & Laurent, G. Multiplication and stimulus invariance in a looming-sensitive neuron. *Journal of Physiology Paris* **98**, 19–34 (2004).
- [51] Gandhi, D., Pinto, L. & Gupta, A. *Learning to fly by crashing* in *IEEE International Conference on Intelligent Robots and Systems 2017-Septe* (2017), 3948–3955. arXiv: 1704.05588.
- [52] Geurten, B. R. H., Kern, R., Braun, E. & Egelhaaf, M. A syntax of hoverfly flight prototypes. *The Journal of experimental biology* **213**, 2461–75 (July 2010).
- [53] Gibson, J. J. The perception of the visual world. *Psychological Bulletin* **48**, 1–259 (1950).
- [54] Giusti, A., Guzzi, J., Ciresan, D. C., He, F.-l., Rodriguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Caro, G. D., Scaramuzza, D. & Gambardella, L. M. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. *IEEE Robotics and Automation Letters* **1**, 661–667 (2016).

-
- [55] Harris, R. A., O'Carroll, D. C. & Laughlin, S. B. Adaptation and the temporal delay filter of fly motion detectors. *Vision Research* **39**, 2603–2613 (1999).
- [56] Hassenstein, B. & Reichardt, W. Systemtheoretische analyse der zeit, reihenfolgen, und vorzeichenbewertung bei der bewegungsperzeption des Rüsselkäfers *Chlorophanus*. *Naturforsch* **11b**, 513–524 (1956).
- [57] Higgins, C. M. Nondirectional motion may underlie insect behavioral dependence on image speed. *Biological Cybernetics* **91**, 326–332 (2004).
- [58] Higgins, C. M., Douglass, J. K. & Strausfeld, N. J. The computational basis of an identified neuronal circuit for elementary motion detection in dipterous insects. *Visual neuroscience* **21**, 567–586 (2004).
- [59] Honegger, D., Meier, L., Tanskanen, P. & Pollefeys, M. An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. *Proceedings - IEEE International Conference on Robotics and Automation*, 1736–1741 (2013).
- [60] Huang, J. V., Wang, Y. & Krapp, H. G. *Wall following in a semi-closed-loop Fly-Robotic interface* in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9793** (2016), 85–96.
- [61] Hyslop, A., Krapp, H. G. & Humbert, J. S. Control theoretic interpretation of directional motion preferences in optic flow processing interneurons. *Biological Cybernetics* **103**, 353–364 (2010).
- [62] Kern, R., Boeddeker, N., Dittmar, L. & Egelhaaf, M. Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information. *Journal of Experimental Biology* **215**, 2501–2514 (July 2012).
- [63] Keshavan, J., Gremillion, G., Escobar-Alvarez, H. & Humbert, J. S. A μ analysis-based, controller-synthesis framework for robust bioinspired visual navigation in less-structured environments. *Bioinspiration and Biomimetics* **9** (2014).
- [64] Koenderink, J. J. & van Doorn, A. J. Facts on optic flow. *Biological Cybernetics* **56**, 247–254 (1987).
- [65] Kraft, P., Evangelista, C., Dacke, M., Labhart, T. & Srinivasan, M. V. Honeybee navigation: Following routes using polarized-light cues. *Philosophical Transactions of the Royal Society B: Biological Sciences* **366**, 703–708 (2011).
- [66] Krapp, H. G. How a fly escapes the reflex trap. *Nature Neuroscience* **18**, 1192–1194 (2015).
- [67] Krapp, H. G. Sensory Integration: Neuronal Adaptations for Robust Visual Self-Motion Estimation. *Current Biology* **19**, R413–R416 (2009).

Bibliography

- [68] Land, M. F. Visual Acuity in Insects. *Annual Review of Entomology* **42**, 147–177 (1997).
- [69] Langelaan, J. & Rock, S. Towards Autonomous UAV Flight in Forests. *Proc of AIAA Guidance Navigation and*, 1–13 (2005).
- [70] Lecoeur, J., Baird, E. & Floreano, D. *A Bee in the Mirror: A Bio-Inspired Model for Vision Based Mid-Air Collision Avoidance (poster)* in *RIN13 Bionav -The application of animal navigation techniques in autonomous vehicles*, Royal Holloway College (2013).
- [71] Lecoeur, J., Baird, E. & Floreano, D. Insect inspired algorithm for joint obstacle avoidance and navigation in forest-like environment. (*in preparation*).
- [72] Lecoeur, J., Baird, E. & Floreano, D. Spatial Encoding of Translational Optic Flow in Planar Scenes by Elementary Motion Detector Arrays. *Scientific Reports* **8**, 5821 (2018).
- [73] Lecoeur, J., Baird, E. & Floreano, D. Visual acuity requirements for mid-air head-on collision avoidance: a scaling law study. (*in preparation*).
- [74] Lecoeur, J., Dacke, M., Floreano, D. & Baird, E. The role of optic flow pooling in insect flight control in cluttered environments. *Scientific Reports in review* (2018).
- [75] Lecoeur, J., Dousse, N., Heitz, G., Schill, F., L'Éplattienier, G. & Huber, B. MAV'RIC: Open source software library to build drone autopilots. (*in preparation*).
- [76] LeCun, Y., Yoshua, B. & Geoffrey, H. Deep learning. *Nature* **521**, 436–444. arXiv: arXiv:1312.6184v5 (2015).
- [77] Lee, A. B., Mumford, D. & Huang, J. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision* **41**, 35–59 (2001).
- [78] Li, J., Lindemann, J. P. & Egelhaaf, M. Peripheral Processing Facilitates Optic Flow-Based Depth Perception. *Frontiers in Computational Neuroscience* **10**, 111 (2016).
- [79] Liang, P., Heitwerth, J., Kern, R., Kurtz, R. & Egelhaaf, M. Object representation and distance encoding in three-dimensional environments by a neural circuit in the visual system of the blowfly. *Journal of Neurophysiology* **107**, 3446–3457 (2012).
- [80] Linander, N., Dacke, M. & Baird, E. Bumblebees measure optic flow for position and speed control flexibly within the frontal visual field. *Journal of Experimental Biology*, 1051–1059 (2015).
- [81] Linander, N., Baird, E. & Dacke, M. Bumblebee flight performance in environments of different proximity. *Journal of Comparative Physiology A* **202**, 97–103 (2016).
- [82] Linander, N., Baird, E. & Dacke, M. How bumblebees use lateral and ventral optic flow cues for position control in environments of different proximity. *Journal of Comparative Physiology A* **203**, 343–351 (2017).

- [83] Lindemann, J. P., Kern, R., van Hateren, J. H., Ritter, H. & Egelhaaf, M. On the Computations Analyzing Natural Optic Flow: Quantitative Model Analysis of the Blowfly Motion Vision Pathway. *Journal of Neuroscience* **25**, 6435–6448 (2005).
- [84] Lindemann, J. P. & Egelhaaf, M. Texture dependence of motion sensing and free flight behavior in blowflies. *Frontiers in behavioral neuroscience* **6**, 92 (Jan. 2012).
- [85] Lindemann, J. *Visual navigation of a virtual blowfly* PhD thesis (2006).
- [86] Longden, K. D. & Krapp, H. G. Sensory neurophysiology: Motion vision during motor action. *Current Biology* **21**, 1684 (2011).
- [87] Loquercio, A., Maqueda, A. I., Del-Blanco, C. R. & Scaramuzza, D. *DroNet: Learning to Fly by Driving* in *IEEE Robotics and Automation Letters* **3** (2018), 1088–1095.
- [88] Lucas, B. D. & Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. *Imaging* **130**, 674–679 (1981).
- [89] Maisak, M. S., Haag, J., Ammer, G., Serbe, E., Meier, M., Leonhardt, A., Schilling, T., Bahl, A., Rubin, G. M., Nern, A., Dickson, B. J., Reiff, D. F., Hopp, E. & Borst, A. A directional tuning map of Drosophila elementary motion detectors. *Nature* **500**, 212–6 (Aug. 2013).
- [90] Meier, L., Honegger, D. & Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. *Proceedings - IEEE International Conference on Robotics and Automation 2015-June*, 6235–6240 (2015).
- [91] Miehlebradt, J. C., Cherpillod, A. T., Mintchev, S., Coscia, M., Artoni, F., Floreano, D. & Micera, S. Data-driven body-machine interface for the accurate control of drones. *Proceedings of the National Academy of Sciences*, 201718648 (2018).
- [92] Neumann, T. & Bulthoff, H. Insect-Inspired Visual Control of Translatory Flight. *Advances in Artificial Life. ECAL 2001. Lecture Notes in Computer Science* **2159**, 627–636 (2001).
- [93] Neumann, T. R. Modeling Insect Compound Eyes: Space-Variant Spherical Vision. *Lncs* **2525**, 360–367 (2002).
- [94] Neumann, T. R. & Bulthoff, H. H. Behavior-oriented vision for biomimetic flight control. *Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics* **203**, 196–203 (2002).
- [95] O’Carroll, D. C., Barnett, P. D. & Nordström, K. Temporal and spatial adaptation of transient responses to local features. *Frontiers in Neural Circuits* **6**, 1–12 (2012).
- [96] Pericet-Camara, R., Bahi-Vila, G., Lecoeur, J. & Floreano, D. *Miniature artificial compound eyes for optic-flow-based robotic navigation* in *2014 13th Workshop on Information Optics, WIO 2014* (2014).

Bibliography

- [97] Plett, J., Bahl, A., Buss, M., Kühnlenz, K. & Borst, A. Bio-inspired visual ego-rotation sensor for MAVs. *Biological Cybernetics* **106**, 51–63 (2012).
- [98] Portelli, G., Serres, J., Ruffier, F. & Franceschini, N. Modelling honeybee visual guidance in a 3-D environment. *Journal of Physiology Paris* **104**, 27–39 (2010).
- [99] Portelli, G., Ruffier, F. & Franceschini, N. Honeybees change their height to restore their optic flow. *Journal of Comparative Physiology A* **196**, 307–313 (2010).
- [100] Portelli, G., Ruffier, F., Roubieu, F. L. & Franceschini, N. Honeybees' speed depends on dorsal as well as lateral, ventral and frontal optic flows. *PLoS ONE* **6** (ed Krapp, H. G.) 10 (2011).
- [101] Portelli, G., Serres, J. R. & Ruffier, F. Altitude control in honeybees: Joint vision-based learning and guidance. *Scientific Reports* **7**, 1–10 (2017).
- [102] Portelli, G., Serres, J., Ruffier, F. & Franceschini, N. A 3D insect-inspired visual autopilot for corridor-following in *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob 2008* (2008), 19–26.
- [103] Raharijaona, T., Serres, J., Vanhoutte, E. & Ruffier, F. Toward an insect-inspired event-based autopilot combining both visual and control events. *2017 3rd International Conference on Event-Based Control, Communication and Signal Processing, EBCCSP 2017 - Proceedings* (2017).
- [104] Reber, T., Vahakainu, a., Baird, E., Weckstrom, M., Warrant, E. & Dacke, M. Effect of light intensity on flight control and temporal properties of photoreceptors in bumblebees. *Journal of Experimental Biology*, 1339–1346 (2015).
- [105] Reiser, M. B. & Dickinson, M. H. A test bed for insect-inspired robotic control. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* **361**, 2267–2285 (Oct. 2003).
- [106] Riley, J. R. & Osborne, J. L. *Flight trajectories of foraging insects: observations using harmonic radar* in *Insect Movement: Mechanisms and Consequences* (2001), 129–157.
- [107] Rognon, C., Mintchev, S., Dell'Agnola, F. I. T., Cherpillod, A. T., Atienza Alonso, D. & Floreano, D. FlyJacket: An Upper Body Soft Exoskeleton for Immersive Drone Control. *IEEE Robotics and Automation Letters* **3**, 2362–2369 (2018).
- [108] Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A. & Hebert, M. Learning monocular reactive UAV control in cluttered natural environments. *Proceedings - IEEE International Conference on Robotics and Automation*, 1765–1772. arXiv: arXiv:1211.1690v1 (2013).

-
- [109] Ruffier, F. & Franceschini, N. OCTAVE, a bioinspired visuo-motor control system for the guidance of Micro-Air-Vehicles. *Bioengineered and Bioinspired Systems* **5119**, 1–12 (2003).
- [110] Ruffier, F. & Franceschini, N. Optic flow regulation: The key to aircraft automatic guidance. *Robotics and Autonomous Systems* **50**, 177–194 (2005).
- [111] Rumelhart, D. E. & Zipser, D. Feature discovery by competitive learning. *Cognitive Science* **9**, 75–112 (1985).
- [112] Scherer, S., Singh, S., Chamberlain, L. & Saripalli, S. Flying fast and low among obstacles. *Proceedings - IEEE International Conference on Robotics and Automation*, 2023–2029 (Apr. 2007).
- [113] Schilling, F., Lecoœur, J., Schiano, F. & Floreano, D. Learning Vision-based Cohesive Flight in Drone Swarms. arXiv: 1809.00543 (2018).
- [114] Schwegmann, A., Lindemann, J. P. & Egelhaaf, M. Depth information in natural environments derived from optic flow by insect motion detection system: a model analysis. *Frontiers in Computational Neuroscience* **8**, 1–15 (2014).
- [115] Schwegmann, A., Lindemann, J. P. & Egelhaaf, M. Temporal statistics of natural image sequences generated by movements with insect flight characteristics. *PLoS ONE* **9** (2014).
- [116] Serres, J., Dray, D., Ruffier, F. & Franceschini, N. A vision-based autopilot for a miniature air vehicle: Joint speed control and lateral obstacle avoidance. *Autonomous Robots* **25**, 103–122 (2008).
- [117] Serres, J. R., Masson, G. P., Ruffier, F. & Franceschini, N. A bee in the corridor: Centering and wall-following. *Naturwissenschaften* **95**, 1181–1187 (Sept. 2008).
- [118] Serres, J. R. & Ruffier, F. Biomimetic Autopilot Based on Minimalistic Motion Vision for Navigating along Corridors Comprising U-shaped and S-shaped Turns. *Journal of Bionic Engineering* **12**, 47–60 (2015).
- [119] Serres, J. R. & Viollet, S. Insect-inspired vision for autonomous vehicles. *Current Opinion in Insect Science* **30**, 46–51 (2018).
- [120] Spaethe, J. & Chittka, L. Interindividual variation of eye optics and single object resolution in bumblebees. *Journal of Experimental Biology* **206**, 3447–3453 (Oct. 2003).
- [121] Srinivasan, M. V., Zhang, S. W. & Chandrashekara, K. Evidence for two distinct movement detecting mechanisms in insect vision. *Naturwissenschaften* **80**, 38–41 (1993).

Bibliography

- [122] Srinivasan, M. V., Lehrer, M., Kirchner, W. H. & Zhang, S. W. Range perception through apparent image speed in freely flying honeybees. *Visual Neuroscience* **6**, 519–535 (1991).
- [123] Srinivasan, M. V., Zhang, S. W. & Lehrer, M. Honeybee navigation: Odometry with monocular input. *Animal Behaviour* **56**, 1245–1259 (Jan. 1998).
- [124] Srinivasan, M. V. & Zhang, S. Visual Motor Computations in Insects. *Annual Review of Neuroscience* **27**, 679–696 (Jan. 2004).
- [125] Srinivasan, M. V., Zhang, S. W., Chahl, J. S., Barth, E. & Venkatesh, S. How honeybees make grazing landings on flat surfaces. *Biological cybernetics* **83**, 171–183 (2000).
- [126] Strübbe, S., Stürzl, W. & Egelhaaf, M. Insect-inspired self-motion estimation with dense flow fields-an adaptive matched filter approach. *PLoS ONE* **10**, e0128413 (2015).
- [127] Strydom, R., Thurrowgood, S. & Srinivasan, M. V. Airborne vision system for the detection of moving objects. *Australasian Conference on Robotics and Automation, ACRA*, 2–4 (2013).
- [128] Sugiura, H. & Dickinson, M. H. The generation of forces and moments during visual-evoked steering maneuvers in flying *Drosophila*. *PLoS ONE* **4**, e4883 (Jan. 2009).
- [129] Taylor, G. J., Luu, T., Ball, D. & Srinivasan, M. V. Vision and air flow combine to streamline flying honeybees. *Scientific reports* **3**, 2614 (2013).
- [130] Thurrowgood, S., Moore, R. J. D. & Srinivasan, M. V. Optic Flow for Aircraft Control (2012).
- [131] Van der Schaaf, A. & Van Hateren, J. H. Modelling the power spectra of natural images: Statistics and information. *Vision Research* **36**, 2759–2770 (1996).
- [132] Van Breugel, F. & Dickinson, M. H. The visual control of landing and obstacle avoidance in the fruit fly *Drosophila melanogaster*. *Journal of Experimental Biology* **215**, 1783–1798 (2012).
- [133] Van Breugel, F., Morgansen, K. & Dickinson, M. H. Monocular distance estimation from optic flow during active landing maneuvers. *Bioinspiration & Biomimetics* **9**, 025002 (2014).
- [134] Wiederman, S. D., Shoemaker, P. A. & O’Carroll, D. C. A model for the detection of moving targets in visual clutter inspired by insect physiology. *PLoS ONE* **3**, e2784 (Jan. 2008).
- [135] Zanker, J. M., Srinivasan, M. V. & Egelhaaf, M. Speed tuning in elementary motion detectors of the correlation type. *Biological cybernetics* **80**, 109–16 (1999).

- [136] Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **8689 LNCS**, 818–833. arXiv: 1311.2901 (2014).
- [137] Zoran, D. & Weiss, Y. Natural Images, Gaussian Mixtures and Dead Leaves. *Advances in Neural Information Processing Systems*, 1736–1744 (2012).
- [138] Zufferey, J., Beyeler, A. & Floreano, D. Near-obstacle flight with small UAVs. *UAV' 2008: International Symposium on Unmanned Aerial Vehicles*, 1–16 (2008).
- [139] Zufferey, J. C. & Floreano, D. *Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control* in *Proceedings - IEEE International Conference on Robotics and Automation* **2005** (2005), 2594–2599.
- [140] Zufferey, J.-c. *Bio-Inspired Vision-Based Flying Robots* PhD thesis (EPFL, 2005), 190.

Julien LECOEUR

Mail: julien.lecoeur@gmail.com

Tel. : (+33) 7 82 38 02 19

10, Place du Marché,

74 200 Thonon-les-Bains

FRANCE



PhD Student in Robotics

Bio-Inspired Drone Design and Visual Flight Control

EDUCATION

Studies

- 2012 – 2018** **DOCTORAL STUDIES IN ROBOTICS**, *École Polytechnique Fédérale de Lausanne*
Insect-Inspired Visual Perception for Flight Control and Collision Avoidance.
Under the supervision of Prof. Dario Floreano and Prof. Emily Baird.
- 2009 – 2012** **MASTER OF SCIENCE AND ENGINEERING**, *École Centrale Lyon*
Majors : Computer science, Embedded electronics.
- 2011 – 2012** **RESEARCH MASTER IN DIGITAL IMAGE PROCESSING**, *University of Lyon 1*
Majors : Feature extraction, Pattern recognition, Augmented reality.
- 2006 – 2009** **PREPARATORY SCHOOL**, *Lycée Victor Grignard*
Intensive preparation course for the national entrance exams to the French "Grandes Écoles".
- 2006** **BACCALAUREAT**, *Lycée Alexis De Tocqueville*
Graduation from high school with high honours.

Languages

- French** Native speaker
- English** Fluent
- Spanish** Basics

TEACHING

- 2018** Lecturer for the course "Aerial Robots", teaching Master-level students on the topic on flying robotics through three lectures on control, navigation and bio-inspiration, and through practical work using the PX4 autopilot.
- 2013 – 2017** Teacher assistant for the course "Mobile Robots", teaching Master-level students the basics of robotics (sensor, control, navigation, etc.) through practical work on ground and flying robots.
- 2013 – 2018** Technical advisor for the teams of a robotic competition involving Master students from multidisciplinary fields.
- 2012 – 2018** Supervisor of more than 30 Master-level students, working on various projects related to my research.

PUBLICATIONS

Lead author

- 2019** LecoEUR, J., Dacke, M., Floreano, D., & Baird, E. (2019). Insect flight control in cluttered environments : the role of spatial pooling of optic flow. *Nature Scientific Reports* (in review).
- 2018** LecoEUR, J., Baird, E., & Floreano, D. (2018). Spatial Encoding of Translational Optic Flow in Planar Scenes by Elementary Motion Detector Arrays. *Nature Scientific Reports*, 8(1), 5821.

Co-author

- 2018** Schilling, F., LecoEUR, J., Schiano, F., & Floreano, D. (2018). Learning Vision-based Cohe-sive Flight in Drone Swarms. <http://arxiv.org/abs/1809.00543>.
- 2014** Pericet-Camara, R., Bahi-Vila, G., LecoEUR, J., & Floreano, D. (2014). Miniature arti-ficial compound eyes for optic-flow-based robotic navigation. In 2014 13th Workshop on Information Optics, WIO 2014.
- 2013** Daler, L., LecoEUR, J., Hahlen, P. B., & Floreano, D. (2013). A flying robot with adaptive morphology for multi-modal locomotion. In *IEEE International Conference on Intelli-gent Robots and Systems* (pp. 1361–1366).

TECHNICAL SKILLS

- Programming** Embedded C and C++ (POSIX, Atmel, ST, Microchip), Low level protocols (UART, SPI, I2C), Scripting and data visualization (Python, Numpy, Pandas, Matplotlib, Jupyter, Matlab), Computer vision and machine learning (OpenCV, Scikit-learn, Keras).
- Software** Operating systems (Ubuntu, Arch Linux, Windows), Version control (git, svn, GitHub, GitLab), Continuous integration (Travis, CircleCI, Jenkins, Docker), Technical writing and presentations (Latex, Google Doc, Office), Drawing (Inkscape, Gimp).
- Sensors** Accelerometers, gyroscopes, magnetometers, barometers, GPS, pitot tubes, optic-flow sensors, range sensors (ultrasonic, infrared).
- Building** Experienced in robot design and manufacturing, especially involving flying platforms, servo motors or brushless motors, propellers, magnetic disturbances, vibrations, etc.

OTHER ACTIVITIES

- RC pilot** Conception, building and piloting of radio controlled aircraft models.
Two times French champion in pattern aerobatic RC flying (F3A).
- Drone enthusiast** Contributor to open source autopilot software projects :
— Member of the development team of the *PX4* autopilot.
— Main developer of the *MAV'RIC* autopilot.
- Musician** Drums player in jazz and funk bands.
Member of the association *The Musical Factory*, producing musical comedy shows every year in Paris.

