

# BIO-INSPIRED VISION-BASED FLYING ROBOTS

THÈSE N° 3194 (2005)

PRÉSENTÉE À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut d'ingénierie des systèmes

SECTION DE MICROTECHNIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

**Jean-Christophe ZUFFEREY**

ingénieur en microtechnique diplômé EPF  
de nationalité suisse et originaire de Chippis (VS)

acceptée sur proposition du jury:

Prof. D. Floreano, directeur de thèse

Prof. N. Franceschini, rapporteur

Prof. J.-D. Nicoud, rapporteur

Prof. R. Siegwart, rapporteur

Lausanne, EPFL  
2005



# Acknowledgements

I owe many thanks to my adviser, Prof. Dario Floreano, who gave me both freedom and support to pursue my project. I am equally indebted to him for having adapted this thesis project in order to combine both of my passions: autonomous robotics and aviation.

I would like to thank Prof. Nicolas Franceschini, Prof. Jean-Daniel Nicoud and Prof. Roland Siegwart for participating in my thesis committee. They all three were at the origin of my enthusiasm for autonomous and fly-inspired robotics.

I also thank the Swiss National Science Foundation (grant nr. 620-58049) and the EPFL for funding my research over the years.

I wish to express my gratitude to several former students: Tancredi Merenda, Matthijs van Leeuwen, Michael Bonani, Antoine Beyeler, Yannick Fournier, Alexis Guanella, and many others, whose work had a significant impact to the development of the software and the robotic platforms used in this thesis.

Many people have contributed to this project. Thanks to Cyril Halter for his initial support in building ultra-light flying robots as well as Dr. Jacques Hertig and Dr. Pascal Goulpié for their assistance during the wind-tunnel experiments. Many thanks to André Guignard, Georges Vaucher and Peter Brühlmeier for their technical support in miniature mechanics and electronics. Thanks are also due to Dominique Etienne and Marie-Jo Pellaud for administrative assistance and advice, to Daniel Burnier, Samir Bouabdallah, André Noth, Alexandre Colot, and Gilles Caprari for sharing many ideas (and chips) related to flying and small robots. I am particularly indebted to Markus Waibel, Antoine Beyeler, Frederic Pont, and Céline Ray for proof-reading and discussing some aspects of my dissertation.

Many people have enriched my academic experience during the past four years: Claudio Mattiussi and his rigorous advices balanced by many surrealist discussions, Dr. Francesco Mondada and his sense of humour and organisation, and all my EPFL colleagues, among which

the ones from the first hour, Jesper Blynel and Daniel Roggen. Many thanks are also due to other scientists for fruitful collaborations, in particular people that I met in Telluride, among which Dr. Jörg Kramer and Dr. Shih-Chii Liu.

I would like to thank all my friends for their patience, my parents, Daniel and Liliane, for nurturing my intellectual interests since the very beginning (and for providing technical assistance during experimental phases of my thesis), my sister Sabine for her support (meals and do-it-yourself ideas), and my brother Julien for his critical insight into my (scientific) work.

Finally, I would like to thank Céline for her love, support, and understanding, and for making everything worthwhile.

# Abstract

There are not yet autonomous flying robots capable of manoeuvring in small cluttered environments as insects do. Encouraged by this observation, this thesis presents the development of ultra-light flying robots and control systems going one step toward fully autonomous indoor aerial navigation. The substantial weight and energy constraints imposed by this indoor flying robots preclude the use of powerful processors and active distance sensors. Moreover, flying systems require fast sensory-motor mapping despite their limited processing power.

In order to cope with those apparently contradictory constraints, our approach takes inspiration from flying insects, which display efficient flight control capability in complex environments in spite of their limited weight and relatively tiny brain. In particular, they are able to stabilise their course, avoid obstacles and control their altitude, which represents the basic mechanisms we want to have on an indoor flying robot.

To achieve efficient flight control, insects rely essentially on two sensory modalities: vision and gyroscope. They possess two low-resolution, compound eyes which are particularly sensitive to image motion (optic flow). In their visual system, some neurons are known to be responsible for detecting self-motion and impending collisions based on optic-flow. Gyroscopic information coming from two mechanosensors located behind the wings complements visual cues in tasks such as gaze and course stabilisation.

In this thesis, we explore the application of such biological principles to develop navigation controllers for indoor flying robots. In particular, we address the problem of how low-resolution vision and gyroscopic information can be mapped into actuator commands in real-time to maintain altitude, stabilise the course and avoid obstacles. As an alternative to hand-crafting control systems based on biological principles, in a second phase, we take inspiration from the evolutionary process that eventually generated those animals and apply artificial evolution

to search for alternative control systems and behaviours that can fit the constraints of indoor flying robots.

Instead of replicating the biomechanics of insect flight, our targeted robotic platform is a fixed-wing airplane capable of flying indoors at very low speed ( $<1.5\text{m/s}$ ). This testbed weights only 30-grams and is equipped with several miniature cameras and a small gyroscope. In order to progress gradually in the task of automating indoor flying robots, two other platforms have been developed, namely a miniature wheeled robot and a small indoor airship. All three robotic platforms feature very similar sensors and electronics in order to facilitate the transfer of software modules and control strategies.

Applying the proposed bio-inspired approach, we succeeded in automating the steering (course stabilisation and obstacle avoidance) of the 30-gram airplane in a square textured arena. Then, using artificial evolution with the airship, we obtained alternative navigation strategies based on the same sensory modalities.

# Version abrégée

Il n'existe pas encore de petit robot autonome capable de voler, à l'instar des insectes, dans des environnements intérieurs encombrés. Cette observation a motivé notre thèse qui se propose de développer des robots volants ultralégers ainsi que leur système de contrôle. Les contraintes énergétiques et les limitations de poids de ces engins volants sont telles que l'utilisation de processeurs puissants et de capteurs de distance actifs est impossible. De plus et en dépit de leur puissance de calcul limitée, les systèmes volants requièrent un traitement rapide de l'information sensorielle pour contrôler leurs actionneurs.

Pour faire face à ces contraintes apparemment contradictoires, notre approche s'inspire directement de la biologie des insectes volants qui sont capables de contrôler leur vol efficacement malgré leur faible poids et leur cerveau microscopique. En particulier, ils sont à même de stabiliser leur trajectoire, d'éviter les obstacles et de contrôler leur altitude. Or, ces mécanismes représentent la base de ce qu'un robot volant devrait être capable de faire.

Pour obtenir un contrôle efficace de leur vol, les insectes font principalement appel à deux modalités sensorielles : la vision et des senseurs gyroscopiques. Ils possèdent deux yeux qui, malgré une résolution très limitée, sont particulièrement sensibles aux mouvements d'image (flux optique). Certains neurones du système visuel sont spécialisés dans la détection du mouvement propre de l'insecte et des collisions imminentes. L'information gyroscopique provenant des petits capteurs mécaniques situés derrière les ailes complète le signal visuel pour stabiliser le regard et la trajectoire de l'insecte.

Dans une première phase, nous explorons l'application de tels principes biologiques dans le développement de systèmes de contrôle pour des robots volants d'intérieur. En particulier, nous nous intéressons à la façon de traduire, en temps réel, les informations visuelles (faible résolution) et gyroscopiques en commandes de moteurs pour maintenir l'altitude, stabiliser la trajectoire et éviter les obstacles. Dans

une seconde phase, au lieu de développer “à la main” les systèmes de contrôle nous nous inspirons du processus évolutionniste qui a généré ces insectes. Un processus d'évolution artificielle est alors utilisé pour générer des systèmes de contrôle efficaces ainsi que des comportements satisfaisant aux contraintes des robots volants d'intérieur.

Plutôt que de répliquer la mécanique de vol complexe des insectes, nous avons choisi de construire des systèmes volants plus simples. La première plateforme robotique utilisée est un avion à ailes fixes capable de voler à très basse vitesse ( $<1.5\text{m/s}$ ). Ce robot équipé de plusieurs caméras miniatures et d'un gyroscope ne pèse que 30 grammes. Dans le but de progresser graduellement vers l'automatisation des robots volants d'intérieur, deux autres plates-formes ont été développées : un robot à roues miniature et un petit dirigeable d'intérieur. Tous les trois systèmes sont équipés de façon similaire en ce qui concerne l'électronique et les capteurs, ceci afin de faciliter le transfert des modules informatiques et des stratégies de contrôle.

En appliquant l'approche bio-inspirée proposée, nous avons tout d'abord réussi à automatiser le pilotage (stabilisation de trajectoire et évitement d'obstacle) de l'avion de 30 grammes dans une arène carrée contenant des textures sur les murs. Ensuite, en utilisant l'évolution artificielle avec le dirigeable, nous avons obtenu d'autres stratégies de navigation basées sur les mêmes modalités sensorielles.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview . . . . .	1
1.1.1	The Problem at Hand . . . . .	1
1.1.2	Original Contribution . . . . .	3
1.1.3	Motivation and Background . . . . .	3
1.2	Related Work . . . . .	7
1.2.1	Ultra-light Flying Devices . . . . .	7
1.2.2	Biomimetic Vision-based Robots . . . . .	9
1.2.3	Evolution of Vision-based Navigation . . . . .	14
1.3	Thesis Contents . . . . .	16
1.3.1	Methods . . . . .	16
1.3.2	Thesis Organisation . . . . .	16
<b>2</b>	<b>Biological Principles of Flying Insects</b>	<b>19</b>
2.1	Inspiring Animals . . . . .	19
2.2	Sensor Suite for Flight Control . . . . .	22
2.2.1	Eyes . . . . .	23
2.2.2	Halteres . . . . .	25
2.2.3	Other Mechanosensors . . . . .	26
2.3	Information Processing . . . . .	27
2.3.1	Optic Lobes . . . . .	28
2.3.2	Local Motion Detection . . . . .	30
2.3.3	Analysis of Motion Fields . . . . .	33
2.4	In-Flight Behaviours . . . . .	39
2.4.1	Attitude Control (ATC) . . . . .	39
2.4.2	Course (and Gaze) Stabilisation (CS) . . . . .	41
2.4.3	Obstacle Avoidance (OA) . . . . .	42
2.4.4	Altitude Control (ALC) . . . . .	43
2.5	Summary . . . . .	44

<b>3</b>	<b>Robotic Platforms and Tools</b>	<b>47</b>
3.1	Platforms . . . . .	48
3.1.1	Miniature Wheeled Robot . . . . .	48
3.1.2	Indoor Airship . . . . .	50
3.1.3	Indoor Airplane . . . . .	53
3.1.4	Comparative Summary of Robotic Platforms . . . . .	58
3.2	Unified Embedded Electronics . . . . .	60
3.2.1	Microcontroller Boards . . . . .	60
3.2.2	Sensors . . . . .	62
3.2.3	Communication . . . . .	65
3.3	Software Tools . . . . .	67
3.3.1	Robot Interface . . . . .	67
3.3.2	Robot Simulator . . . . .	68
3.4	Experimental Environments . . . . .	69
3.5	Summary and Discussion . . . . .	72
<b>4</b>	<b>Optic Flow</b>	<b>75</b>
4.1	What is Optic Flow? . . . . .	76
4.1.1	Motion Field and Optic Flow . . . . .	76
4.1.2	Formal Description and Properties . . . . .	77
4.1.3	Motion Parallax . . . . .	80
4.2	Optic Flow Detection . . . . .	81
4.2.1	Issues with Elementary Motion Detectors . . . . .	81
4.2.2	Gradient-based Methods . . . . .	82
4.2.3	Simplified Image Interpolation Algorithm . . . . .	84
4.2.4	Algorithm Assessment . . . . .	86
4.2.5	Implementation Issues . . . . .	89
4.3	Summary . . . . .	92
<b>5</b>	<b>Bio-inspired Navigation Control</b>	<b>93</b>
5.1	Characteristic Patterns of Optic Flow . . . . .	93
5.1.1	Frontal Approach . . . . .	94
5.1.2	Flying Over . . . . .	99
5.2	Information Processing and Control . . . . .	101
5.2.1	Steering Control (CS & OA) . . . . .	102
5.2.2	Altitude Control (ALC) . . . . .	105
5.2.3	Rotatory Optic Flow Cancellation . . . . .	106
5.3	Experiments and Results . . . . .	107
5.3.1	Altitude Control as Wall Following on Wheels . . . . .	108
5.3.2	Steering Control on Wheels . . . . .	109

---

5.3.3 Steering Control of the Airplane . . . . .	110
5.4 Summary and Discussion . . . . .	115
<b>6 Evolutionary Approach</b>	<b>121</b>
6.1 Method . . . . .	122
6.1.1 Rationale . . . . .	122
6.1.2 Evolutionary Process . . . . .	124
6.1.3 Neural Controller . . . . .	125
6.1.4 Fitness Function . . . . .	126
6.2 Experiments on Wheels . . . . .	127
6.2.1 Raw Vision versus Optic Flow . . . . .	127
6.2.2 Coping with Critical Situations . . . . .	132
6.3 Aerial Experiments . . . . .	134
6.3.1 Evolution in Simulation . . . . .	136
6.3.2 Transfer to Reality . . . . .	140
6.4 Summary and Discussion . . . . .	141
<b>7 Concluding Remarks</b>	<b>147</b>
7.1 Main Achievements . . . . .	147
7.2 Potential Applications . . . . .	148
7.3 Future Directions . . . . .	149
<b>A Dynamic Modelling of Indoor Airships</b>	<b>151</b>
A.1 Restoring Forces . . . . .	153
A.2 Propulsion . . . . .	154
A.3 Damping . . . . .	154
A.4 Inertia . . . . .	156
A.5 Coriolis and Centripetal Effects . . . . .	162
<b>B Embedded Neural Network</b>	<b>165</b>
B.1 PIC-compatible Neural Network (PIC-NN) . . . . .	165
B.2 Remote and Embedded Modes . . . . .	166
<b>Bibliography</b>	<b>169</b>
<b>Curriculum Vitae</b>	<b>189</b>



# Chapter 1

## Introduction

As natural selection is inherently opportunistic, the neurobiologist must adopt the attitude of the engineer, who is concerned not so much with analyzing the world than with designing a system that fulfils a particular purpose.

---

R. Wehner (1987)

### 1.1 Overview

#### 1.1.1 The Problem at Hand

There are not yet autonomous flying robots capable of manoeuvring in small cluttered environments as insects do. Driven by this observation, this thesis presents the development of ultra-light flying robots and control systems going one step toward fully autonomous indoor aerial navigation. The substantial weight and energy constraints imposed by indoor flying robots preclude the use of powerful processors and active distance sensors such as sonar or laser range finders (Siegwart and Nourbakhsh, 2004). Moreover, flying systems tend to move faster than terrestrial vehicles of the same size. They thus require fast sensory-motor mapping despite their limited processing power. In order to cope with those apparently contradictory constraints, our approach takes inspiration from *flying insects*, which display efficient flight control capability in complex environments in spite of their limited weight and relatively tiny brain.

Biological inspiration takes place at different levels. The choice of sensory modalities (*low-resolution vision* and *gyroscopic sensors*) is based on the anatomy of flies. At the control level, two different approaches are explored. The first approach consists in mimicking flying

insects in their way of *processing information* and *behaving* using *optic flow*<sup>1</sup>. The second approach relies on *artificial evolution* to automatically develop *neuromorphic controllers* that map sensory signals into motor commands to produce adapted behaviours.

When tackling the realisation of bio-inspired flying robots, not only the physical platforms need to be developed, but the type of behaviours they should display must be decided as well as the environment in which they will be tested. Since in the most general terms the research domain is enormous and the technological challenges of building autonomous indoor flying systems are significant, the scope of this research has deliberately been restricted as follows.

We do not attempt to reproduce the bio-mechanical principles of insect flight, which is in itself a considerable domain of research (Dickinson et al., 1999; Dudley, 2000; Lehmann, 2004). Instead, more classical flight principles are adapted to the constraints of flying indoor. The first step of this project thus consists of building flying platforms able to manoeuvre within limited space, while still having enough lift capability to support the required sensors and electronics. Two flying platforms, a 120cm-long *indoor airship* (also called *blimp*) and a 30-gram *airplane*, have been developed to serve as experimental testbeds and demonstrators. An additional *wheeled robot* featuring similar electronics is also used to first assess new control ideas before transferring them to aerial robots.

At the behavioural level, instead of tackling an endless list of high-level tasks like goal-directed navigation, homing, food seeking, landing, etc., which themselves constitute open research topics even on terrestrial robots, a set of requirements for basic navigation is identified and studied. The behaviour which is at the core of the dissertation can be summarised as "maximising forward translation". This simple behaviour requires a series of more basic mechanisms such as *attitude control* (ATC), *course stabilisation* (CS), *obstacle avoidance* (OA), and *altitude control* (ALC).

With regard to the choice of experimental environments, simple geometries and textures have been chosen in order to allow easy characterisation and manipulation of environmental features. Experimental environments are square rooms with randomly distributed black and white gratings painted on walls. The size of the arenas is adapted to each robot dynamics, e.g., the arena used for experiments with the air-

---

<sup>1</sup>The optic flow is the image motion as projected onto the retina of a visual system. See chapter 4 for details.

plane flying at about 1.5m/s is three times larger (15 by 15m) than the arena for the blimp (5 by 5m), which has a nominal velocity around 50cm/s.

### 1.1.2 Original Contribution

This thesis sits at the intersection of several scientific disciplines such as biology, aerodynamics, micro-engineering, micro-electronics, computer vision, and robotics. One of the main challenges of this work relies in the integration of the knowledge from all these different fields into functional systems able to fly indoor and the proposal of control strategies suitable for such autonomous aircrafts. More specifically, this thesis contributes with:

- the development of indoor flying aircrafts, in particular a 30-gram robotic airplane fitted with two 1D cameras and a small gyroscope (chapter 3),
- the implementation of an optic-flow processing algorithm (chapter 4) as well as an artificial neural network (appendix B) for vision-based navigation in the embedded 8-bit microcontroller,
- a control strategy for vision-based steering and obstacle avoidance of the 30-gram robotic airplane (chapter 5),
- the application of evolutionary algorithms to the control of a physical vision-based flying robot (chapter 6).

### 1.1.3 Motivation and Background

#### Indoor Flying Robots

In the field of mobile robotics, a lot of research has been carried out on wheeled robots moving on flat surfaces. More recently, the types of platforms have expanded to rough-terrain vehicles, humanoid and walking robots. Although some noticeable exceptions exist (see, among others, the ones cited in the next section), aerial systems are under-represented in the list of platforms commonly used in autonomous robotics. This is mainly due to the three following reasons. First, aerial systems are usually costly and require expert backup pilots to secure every experiment. Second, they require a wide clear space that is not always available in the vicinity of the research centers. Moreover, there

are legal restrictions to unmanned aerial vehicles (UAVs) in most parts of the airspace. Third, the military domain represents the main field of applications of UAVs today, and although some civilian opportunities exist in traffic surveillance, search and rescue, visual reconnaissance, weather monitoring, or communication relay, there is not yet a clear commercial interest for truly autonomous aerial robots.

Recently, flying in small indoor environments became possible thanks to technological advances in battery (increase in power to weight ratio) and miniaturisation of electrical motors (Nicoud and Zuferey, 2002; Bouabdallah et al., 2005). This new possibility of flying indoor opens new horizons to roboticists because it dramatically reduces costs and security issues, while releasing researchers from requesting special authorisations from air-law instances. The typical characteristics of flying platforms such as their ability to manoeuvre in three dimensions, the absence of mechanical contact with a reference frame, their complex dynamics and rapid motion represent as many interesting research topics for roboticists. Simultaneously, flying indoor imposes strong constraints toward efficient system integration, minimal weight and low energy consumption. In addition to underwater, space, and aerial robotics (see chapter 7 for further discussion), advances in such issues can contribute to other domains where low-cost, small-size, low-energy vision could be useful for real-time collision detection or other purposes.

### **Inspiration from Biology**

One of the fundamental challenges facing the field of robotics is the design of robust controllers that map sensory input to action. The problem is even more severe when it comes to robots that are small, lightweight and relatively fast, which is the case of indoor flying systems. Engineers have been able to master amazing technologies for flying at very high speed or travelling into space. However, biological information processing systems far outperform today's computers at tasks involving real-time perception in uncontrolled environments, in particular if we take energy efficiency into account. Based on this observation, this thesis aims at understanding what kind of biological principles might be amenable to artificial implementation in order to synthesise systems that typically require energy efficiency, low-power processing and feature complex dynamics.

In this dissertation, the notion of "biological principle" is taken



in a broad meaning, ranging from individual biological features like anatomy of perceptive organs, models of information processing or behaviours, to the evolutionary process at the level of the species. The idea of applying such biological principles to autonomous robots is drawn from the fields of biorobotics<sup>2</sup> (Chang and Gaudiano, 2000; Webb and Consi, 2001) and evolutionary robotics (Nolfi and Floreano, 2000), which is in some sense a sub-domain of biorobotics. These trends have themselves largely been inspired by the new artificial intelligence (new AI) first advocated by Brooks in the early 80s (for a review, see Brooks, 1999) and by the seminal ideas by Braitenberg (1984). When taking inspiration from biological principles, care must be taken to avoid the pitfall of doing biomimicry for itself while forgetting the primary goal of constructing functional autonomous robots. For instance, it would not make sense to replace efficient engineered systems or subsystems by poorly performing bio-inspired systems for the only reason that they are bio-inspired.

One of the main goals of this thesis is to select and apply biological principles to the design of autonomously flying robots because such principles proved to be successful at solving the same kind of problems in biological systems. Although our primary goal is not to assess the biological principles, the fact that they permit efficient results in artificial systems might contribute to the field of biology in the sense that the implemented model is at least a functional explanation of the underlying biological principle. The increasing number of biologists willing to construct mobile robots in order to test their models in artificial systems is indeed remarkable (see, for instance, Srinivasan et al., 1997; Duchon et al., 1998; Lambrinos et al., 2000; Rind, 2002; Reiser and Dickinson, 2003; Franceschini, 2004; Webb et al., 2004). As a result, biorobotics can be seen as a link between the worlds of biology and engineering, in which biorobots (or bio-inspired robots) are used as tools for biologists studying animal behaviour and as testbeds for the study and evaluation of biological algorithms for potential applications to engineering. Furthermore biorobotics can also help biologists to identify and investigate worthwhile issues related to sensory-motor control that could, in turn, provide new solutions to roboticists.

---

<sup>2</sup>Also called bio-inspired robotics or biomimetic robotics.

## Vision as Main Sensory Modality

The weight and energy constraints of indoor flying systems drastically restrict the choice of sensors that can be embedded. Recent developments in MEMS<sup>3</sup> technology allow to measure strength, pressure, magnetic or inertial forces with ultra-light devices weighing only a few milligrams. For what concerns distance perception, the only passive sensory modality that can provide useful information is vision. Active range finders such as laser, infrared or sonar have significant drawbacks such as their inherent weight (they require an emitter and a receiver), their need to send energy into the environment, and their inability to “see” a wide portion of the surroundings unless they are mounted on a scanning system. On the other hand, visual sensors can be made really tiny, do not need to send signals into the environment, and have a larger field of view.

The same considerations have probably driven evolution of flying insects toward the same choice of using vision instead of active range finders to control their flight, avoid obstacles and navigate in complex environments. The main problem of vision is the non-trivial relationship between the raw signal coming from the sensor and the corresponding 3D layout of the surroundings. The mainstream approach to computer vision based on a sequence of pre-processing, segmentation, object extraction, and pattern recognition of each single image is probably not tractable for the embedded processor of an ultra-light flying robot that must respond very quickly in its environments. By taking inspiration from flying insects, this thesis aims at discovering how to map simple visual patterns directly to actuator commands. This is possible because vision is taken as part of a complete agent that must accomplish a task in a given environment (Pfeifer and Lambrinos, 2000). This idea is very close to the ecological approach to visual perception, first developed by Gibson (1950, 1979) and further advocated by Duchon et al. (1998):

Ecological psychology [...] views animals and their environments as “inseparable pairs” that should be described at a scale relevant to the animal’s behavior. So, for example, animals perceive the layout of surfaces (not the coordinates of points in space) and what the layout affords for action (not merely its three-dimensional structure). A main tenet of

---

<sup>3</sup>Micro-Electro-Mechanical Systems

the ecological approach is that the optic array, the pattern of light reflected from these surfaces, provides adequate information for controlling behavior without further inferential processing or model construction. This view is called *direct perception*: The animal has direct knowledge of, and relationship to, its environment as a result of natural laws.

Following this philosophy, no attempt shall be made to explicitly estimate distances separating the artificial eye of our flying robots and the environment. Rather, simple biological models are used to link image motion to actions such as obstacle avoidance or altitude control, without going through complex sequences of image processing.

## 1.2 Related Work

In this section, we first review ultra-light flying systems that are not (yet) autonomous. Then, bio-inspired vision-based robots of different kinds are presented.

### 1.2.1 Ultra-light Flying Devices

This subsection is a review of recent efforts in building micro-mechanical devices that can fly in restricted environments. Lighter-than-air (blimp) platforms (e.g., Zhang and Ostrowski, 1998; Planta et al., 2002; van der Zwaan et al., 2002; Melhuish and Welsby, 2002; da Silva Metelo and Garcia Campos, 2003; Iida, 2003) are not described because their realisation is not technically challenging. Micro air vehicles (MAVs; see for example Mueller, 2001; Grasmeyer and Keennon, 2001; Ettinger et al., 2003) are not tackled either because they are not intended for slow flight in restricted areas. MAVs (as defined by DARPA<sup>4</sup>) fly at around 15m/s, whereas indoor aircrafts should be able to fly below 2m/s in order to steer in typical indoor environments such as offices or houses. Finally, only very small rotorcrafts are described because this type of machines become very risky to fly indoors as soon as their weight increase.

A team at Caltech in collaboration with Aerovironment<sup>TM</sup> developed the first remote-controlled, battery-powered, flapping-wing micro aircraft (Pornsirak et al., 2001). This 20cm-wingspan, 12-gram device (figure 1.1a) has an autonomy of 6 minutes when powered with a

<sup>4</sup>The American Defense Advanced Research Projects Agency.

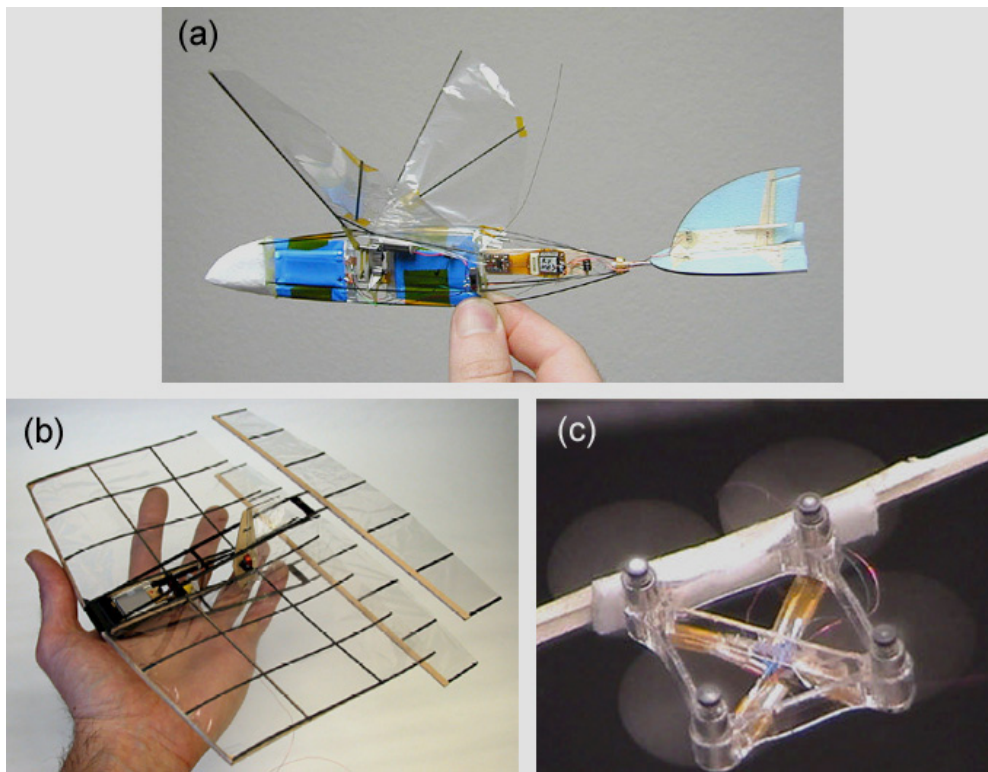


Figure 1.1: (a) Caltech and Aerovironment's 12-gram MicroBat. Reprinted from <http://touch.caltech.edu/research/bat/bat.html>. (b) The 14-gram biplane flapping thruster. Reprinted from Jones et al. (2004). (c) A tethered version of the Mesicopter with 15mm propellers. Reprinted from <http://aero.stanford.edu/mesicopter/>.

lithium-polymer battery. More recently, Jones et al. (2004) engineered a small radio-controlled device propelled by a biplane configuration of flapping wings that move up and down in counter-phase (figure 1.1b). The 14-gram model has demonstrated stable flight at speeds between 2 and 5m/s. Another team at Stanford University (Kroo and Kunz, 2001) proposed a centimeter scale rotorcraft (figure 1.1c), based on four miniature motors with 15mm propellers. However, experiments on lift and stability were done on larger models. Although those flying devices constitute remarkable micro-mechatronic developments, none of them is capable of autonomous navigation, nor has on-board sensors of any type.

In an even smaller size, Ron Fearing's team is attempting to create a micro flying robot (figure 1.2) that replicates wing mechanics and dynamics of flies (Fearing et al., 2000). The planned weight of the final device is approximately 100mg for a 25mm wingspan. Piezoelectric actuators are used for flapping and rotating the wings at about

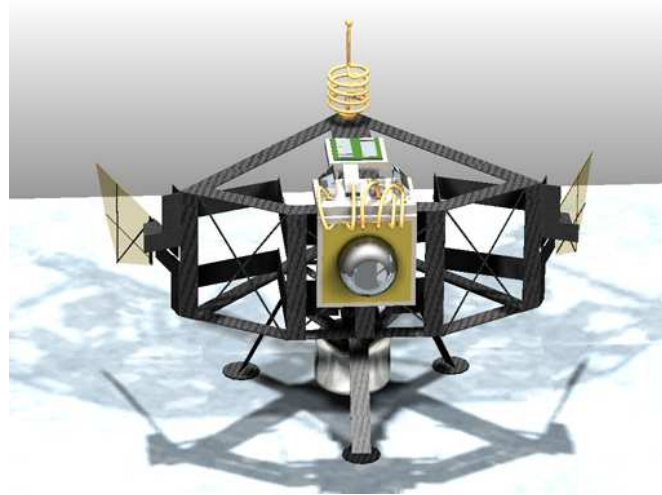


Figure 1.2: Artist's conception of future micromechanical flying insect (MFI). Reprinted from <http://robotics.eecs.berkeley.edu/~ronf/mfi.html>.

150Hz. Electrical power should be supplied from lithium-polymer batteries charged by three miniature solar panels. So far, a single wing on a test rig has generated about 0.5mN average lift while linked to an off-board power supply (Avadhanula et al., 2003). The team is currently working on a bio-mimetic sensor suite for attitude control (Wu et al., 2003), but no test in flight has been reported so far.

### 1.2.2 Biomimetic Vision-based Robots

In the early 90s, research on biomimetic vision-based navigation was mainly carried out on wheeled robots. Although, several functions such as searching, aiming, topological navigation were studied (Franz and Mallot, 2000), this review focuses on the basic behaviour of obstacle avoidance that is related to the contribution of this thesis. More recent work is also reported, which tackles vision-based aerial guidance from a biomimetic perspective. A common aspect of these robots is that they use optic flow as main sensory input for controlling their motion.

#### **Wheeled Robots**

Several models of visual guidance of flying insects have been tested on terrestrial robots. Nicolas Franceschini and his team at CNRS in Marseilles (for a review, see Franceschini, 2004) have spent several years studying the morphological and neurological aspects of the visual system of the fly and its way of detecting optic-flow (see chapter 2). In

order to test their hypothesis on how the fly could use optic-flow, they built an analog electronic circuit modeled upon the neural circuitry of the fly brain and interfaced it with a circular array of photoreceptors on a 12-kg wheeled robot (figure 1.3a). The so-called “robot mouche” was capable of approaching a goal while avoiding obstacles (characterised by higher contrast with respect to a background) on its way (Pichon et al., 1990; Franceschini et al., 1992).

Although some preliminary results in vision-based obstacle avoidance were obtained with a gantry robot by Nelson and Aloimonos (1989), most of the work on biomimetic, vision-based, wheeled robots followed the realisation of the “robot mouche”. Another key player in this domain was Srinivasan and his team at the Australian National University in Canberra. They performed an extensive set of experiments to understand visual performance of honeybees and tested the resulting models on robots (for reviews, see Srinivasan et al., 1997, 1998). For example, they showed that honeybees regulate their direction of flight by balancing the speeds of image motion on the two eyes (Srinivasan et al., 1996). This mechanism was then demonstrated on a wheeled robot equipped with a camera and two mirrors (figure 1.3b) that captured images of the lateral walls and transmitted them to a desktop computer where an algorithm attempted to balance the optic-flow in the two lateral views by steering the robot accordingly (Weber et al., 1997). In the same research team, Sobey (1994) implemented an algorithm inspired by insect flight to drive a vision-based robot (figure 1.3c) in cluttered environments. The algorithm related the position of the camera, the speed of the robot, and the measured optic-flow during translatory motions in order to judge distances from objects and steering accordingly.

Several other groups used insect visual-control systems as models for wheeled robots, either for obstacle avoidance in cluttered environments (Duchon and Warren, 1994; Lewis, 1998) or for corridor following (Coombs et al., 1995; Santos-Victor et al., 1995). Some of these robots used active camera mechanisms for stabilising their gaze in order to cancel spurious optic-flow introduced by self-rotation (see chapter 4). All of the reviewed wheeled robots relied on the fact that they were in contact with a flat surface in order to infer or control their self-motion through wheel encoders. Since flying robots have no contact with ground, those approaches cannot be applied to our project without modifications (for instance the integration of other sensors). Furthermore, the tight weight budget precludes active camera mecha-

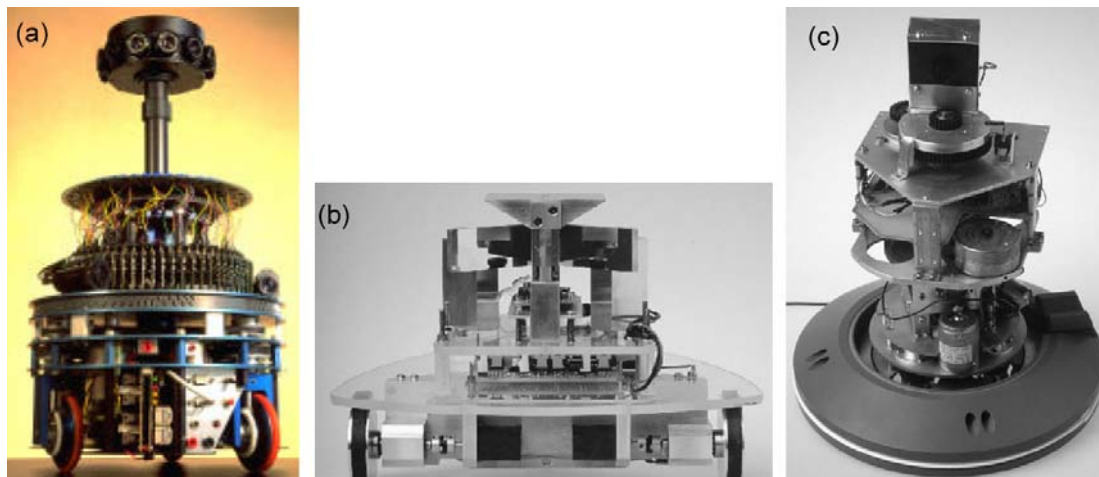


Figure 1.3: (a) “Robot mouche” by Franceschini’s team. Reprinted from Franceschini (2003). (b) Corridor-following robot by Srinivasan’s team. Reprinted from Srinivasan et al. (1998). (c) Obstacle-avoiding robot by Srinivasan’s team. Reprinted from Srinivasan et al. (1998).

nisms for gaze stabilisation. Finally, all the above mentioned robots, except the “robot mouche”, featured off-board processing of 2D images, which is not the goal of our project that aims at self-contained autonomous robots.

### Aerial Robots

A few optic-flow based aerial experiments have been carried out on indoor airships (blimps), although they were not aimed at obstacle avoidance. Iida and colleagues demonstrated visual odometry and course stabilisation (Iida and Lambrinos, 2000; Iida, 2001, 2003) using such a platform equipped with an omnidirectional camera (figure 1.4a-b) down-streaming images to an off-board computer for optic-flow estimation. Planta et al. (2002) presented a blimp using an off-board neural controller for course and altitude stabilisation in a rectangular arena equipped with regular checkerboard patterns. However, altitude control produced poor results.

Specific studies on altitude control have been conducted by Franceschini’s team, first in simulation (Mura and Franceschini, 1994), and more recently with tethered helicopters (figure 1.4c; Netter and Franceschini, 2002; Ruffier and Franceschini, 2004). Although the control was done off-board for analysis purpose, the viability of regulating altitude of a small helicopter by using the amount of ventral

optic flow as detected by a minimalist vision system composed of only 2 photoreceptors was demonstrated. The regulation system even did not need to know the velocity of the aircraft. Since these helicopters were tethered, the degrees of freedom were deliberately limited to 3 and the pitch angle could directly be controlled by means of a servomotor mounted at the articulation between the boom and the aircraft. The knowledge of the absolute pitch angle allowed to ensure the vertical orientation of the optic-flow detector (active gaze stabilisation) when the rotorcraft was tilted fore and aft to modulate its velocity.

In an attempt of using optic-flow to control altitude of a free-flying UAV, Chahl et al. (2004) took inspiration from the honeybee's landing strategy (Srinivasan et al., 2000) to regulate the pitch angle using ventral optic-flow during descent. However, real world experiments produced very limited results, mainly because of the spurious optic-flow introduced by corrective pitching movements.<sup>5</sup>

In order to test a model of obstacle avoidance in flies (Tammero and Dickinson, 2002b), Reiser and Dickinson (2003) set up an experiment with a robotic gantry emulating the fly's motion in a randomly textured circular arena. This experiment successfully demonstrated robust visual steering based on biological models. Another significant body of work in simulation (Neumann and Bühlhoff, 2001, 2002) demonstrated full 3D, vision-based navigation (attitude control based on light intensity gradient; course stabilisation, obstacle avoidance and altitude control based on optic-flow). However, the dynamics of the simulated agent was minimalist (not representative of a real flying robot) and the environment featured a well-defined light intensity gradient, which might not always be available in real-world conditions. Very recently, Muratet et al. (2005) developed an efficient control strategy for obstacle avoidance with a simulated helicopter flying in urban canyons. In some respects, this strategy based on a combination of inertial and visual information is very close to what is proposed in this thesis (chapter 5). However, this work in simulation relied on a full-featured autopilot (with GPS, inertial measurement unit, and altitude sensor) as low-level flight controller and made use of a relatively high resolution camera. Those components are likely to be too heavy in reality for mounting them on an ultra-light indoor aircraft (which was not the purpose of the above researchers).

The attempts at automating real free-flying UAVs using bio-inspired

---

<sup>5</sup>See the end of section 5 in Chahl et al. (2004). See also Ruffier (2004) for other points of criticism.



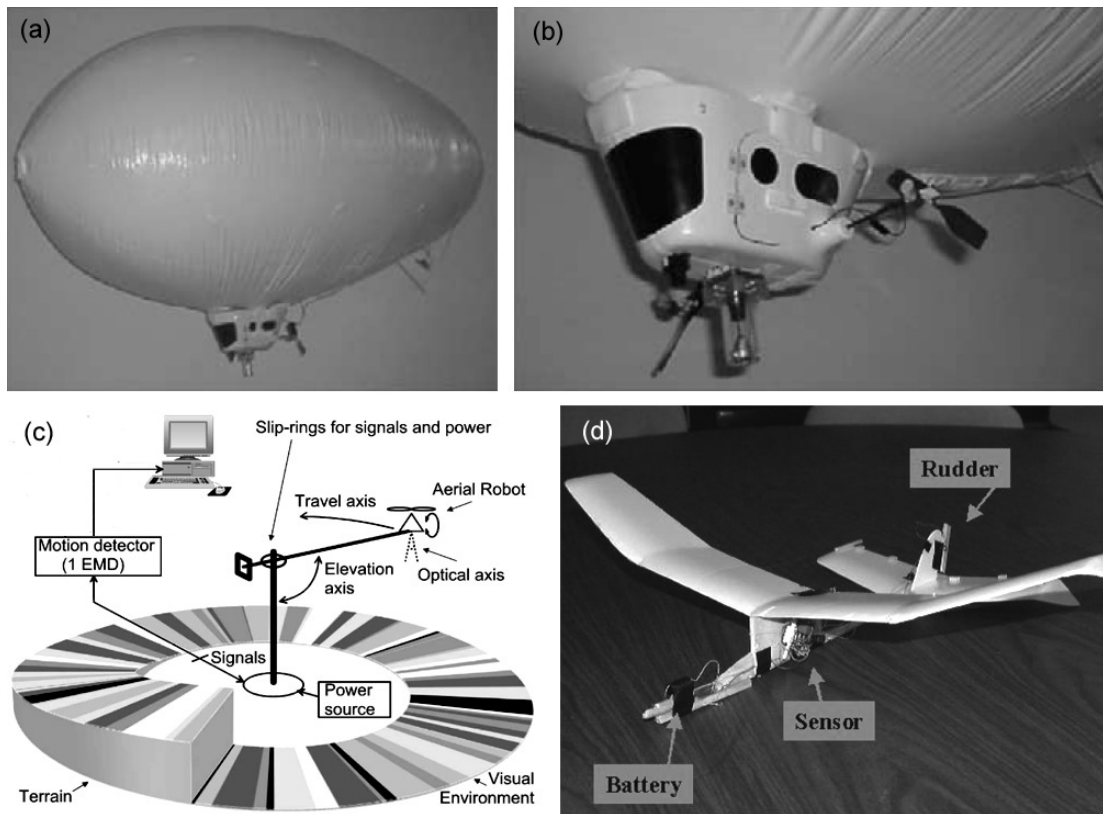


Figure 1.4: (a-b) The blimp equipped with an omnidirectional vision system. Reprinted from Iida (2003). (c) Outline of the tethered helicopter for altitude control study. Adapted from Ruffier and Franceschini (2004). (d) Glider with a lateral optic-flow detector. Reprinted from Barrows et al. (2001).

vision are quite limited. Barrows et al. (2001) reported preliminary experiments on lateral obstacle avoidance in a gymnastic hall with a model glider (figure 1.4d) carrying a 25g optic-flow sensor. Although no data supporting the described results are provided, it seems that the glider could steer away from a wall when tossed toward it at a shallow angle. A further experiment with a 1m wingspan aircraft (see picture in Barrows et al., 2002) were performed outdoor. The purpose was essentially to demonstrate altitude control with a ventral optic-flow sensor. A simple (on/off) altitude control law managed to maintain the aircraft airborne for 15 minutes, during which 3 failures occurred where the human pilot had to rescue the aircraft because it dropped too close to the ground. Very recently, Green et al. (2004) carried out an experiment of lateral obstacle avoidance with an indoor aircraft<sup>6</sup> equipped with a laterally-mounted 4.8g optic-flow sensor. Only one

<sup>6</sup>Overall weight of this airplane with electronics and visual system is not indicated.

trial, in which the aircraft avoids a basketball net is described and illustrated with video screen-shots. Since only one sensor was used, the aircraft could detect obstacles only on one side.

Although these real-world experiments by Barrows, Green and colleagues are remarkable, no continuous autonomous flight with obstacle avoidance has been reported so far. Also lacking is a detailed analysis based on flight data. Furthermore, no specific attention has been made to spurious optic-flow signals introduced by rotations of the aircraft. The authors assumed, more or less implicitly, that rotational components of optic flow arising from changes in aircraft orientation are smaller than the translational component. However, this assumption usually does not hold and the issue deserves more careful attention. Finally, no frontal obstacle avoidance experiments have been described so far.

### **1.2.3 Evolution of Vision-based Navigation**

Instead of hand-crafting robot controllers based on biological principles, an alternative approach consists in using genetic algorithms<sup>7</sup> (GAs). When applied to the design of robot controllers, this method is called evolutionary robotics (ER) and goes as follows (Nolfi and Floreano, 2000):

An initial population of different artificial chromosomes, each encoding the control system (and sometimes the morphology) of a robot, are randomly created and put in the environment. Each robot (physical or simulated) is then let free to act (move, look around, manipulate) according to a genetically specified controller while its performance on various tasks is automatically evaluated. The fittest robots are allowed to reproduce by generating copies of their genotypes with the addition of changes introduced by some genetic operators (e.g., mutations, crossover, duplication). This process is repeated for a number of generations until an individual is born which satisfies the performance criterion (fitness function) set by the experimenter.

Some experiments in ER have already demonstrated successful results at evolving vision-based robots to navigate. Those related to obstacle avoidance are briefly reviewed in this section.

---

<sup>7</sup>Search procedure based on the mechanisms of natural selection (Goldberg, 1989).

At Max-Plank Institute in Tübingen, Huber et al. (1996) carried out a set of experiments where simulated agents evolved their visual sensor orientations and sensory-motor coupling. The task of the agent was to navigate as far as possible in a simulated corridor-like environment with a few perpendicular obstacles. Four photodetectors were brought together to compose two elementary motion detectors (see chapter 2), one on each side of the agent. The simple sensory-motor architecture was inspired from Braitenberg (1984). Despite their minimalist sensory system, the autonomous agents successfully adapted to the task during artificial evolution. Best evolved individuals sensor orientation and sensory-motor coupling suitable for obstacle avoidance. Going one step further, Neumann et al. (1997) showed that the same approach can be applied to aerial simulated agents. The minimalist flying system was equipped with two horizontal and two vertical elementary motion detectors and evolved in the same kind of textured corridor. Although the agents developed effective behaviours to avoid horizontal and vertical obstacles, those results are of limited interest when it comes to physical flying robots because the simulated agents featured very basic dynamics and had no freedom around pitch and roll axis. Moreover, the visual input was probably too ideal to be representative of real world conditions.<sup>8</sup>

In our team, experiments have been carried out where a small Khepera robot (see chapter 3) evolved the ability to navigate in a randomly textured environment (Floreano and Mattiussi, 2001). The robot was equipped with a 1D camera of 16 pixels with a 36° field-of-view as only sensor. Evolution found relatively quickly functional neuromorphic controllers capable of navigating in the environment without hitting walls using a very simple genetic encoding and fitness function. Note that unlike the experiments by Huber and Neumann, this approach did not rely on optic flow. The visual input was only pre-processed with a spatial high-pass filter before feeding a general purpose neural network. In that case, the sensory morphology was not concurrently evolved with the controller architecture. In this dissertation, we demonstrate the application of the same approach to the navigation of physical flying robots.

---

<sup>8</sup>Other authors evolved terrestrial vision-based robots in simulation (for example, Cliff and Miller, 1996; Cliff et al., 1997), but the chosen tasks (pursuit and evasion) are not directly related to the ones tackled in this thesis. The same team also worked with a gantry robot for real-world visually-guided behaviours like shape discrimination (Harvey et al., 1994).

In another set of experiments (Marocco and Floreano, 2002; Floreano et al., 2004), both in simulation and with a real robot, explored the evolution of active visual mechanisms allowing evolved controllers to decide where to look while they were navigating in their environment. Although those experiments yielded interesting results, this approach is discarded for our application because an active camera mechanism is too heavy for our lightweight aerial robots.

## **1.3 Thesis Contents**

### **1.3.1 Methods**

Our approach consists in reviewing the main biological principles enabling flying insects to control their flight. The principles that seems transferable and useful in the design of autonomous flying devices are first assessed on a wheeled robot featuring the same sensors and processor as the flying platforms. When good results are obtained, algorithms are then transferred and adapted to aerial platforms.

The experiments presented in this thesis can be divided into two groups. In the first one, biological principles are used to hand-craft control systems that drive the robots. In the second group, the biological principle of evolution is employed to automatically develop the robots' controllers.

### **1.3.2 Thesis Organisation**

Chapter 2: *Biological Principles of Flying Insects*. As we are interested in taking inspiration from flying insects, this chapter reviews biological principles, from sensor anatomy to information processing and behaviour, that may be amenable to artificial implementation. This is not a comprehensive biological description of flying insects, but rather a pragmatic insight into selected topics from an engineering perspective.

Chapter 3: *Robotic Platforms and Tools*. The platforms and tools that have been developed in order to test the proposed approach are introduced in this chapter. The targeted robotic platform consists of a 30-gram indoor aircraft. Autonomous steering of this ultra-light flying robot will be demonstrated in chapter 5. However, two other robots have been employed featuring intermediate dynamic complexity. The first one is a miniature wheeled robot allowing for rapid assessment

of new ideas. The second one is an indoor airship (blimp) constituting a simpler and more robust aerial platform, hence enabling more risky approaches, such as artificial evolution presented in chapter 6. An overview is given of the three robots, their very similar electronic equipment and sensors as well as their dynamic properties. The experimental environments adapted to the size of each robot are also described. Additionally, the software tools allowing for interfacing and simulation of the robots are briefly presented.

Chapter 4: *Optic Flow*. As will be explained in chapter 2, the detection of visual motion information plays a prominent role in the behaviours of flying insects. This chapter is thus devoted to optic flow, its formal definition, its properties, and its detection. Taking into account the very limited processing power available on-board our lightweight robots, an efficient algorithm for estimating optic flow is proposed and tested in real-world conditions.

Chapter 5: *Bio-inspired Navigation Control*. Taking inspiration from the models and principles described in chapter 2 and fitting the constraints imposed by the body of the robots presented in chapter 3, we implement basic visually-guided behaviours using optic-flow detection introduced in chapter 4. Obstacle avoidance and altitude control are first demonstrated on wheels. Then the control system for obstacle avoidance is augmented with a course stabilisation mechanism and transferred to the ultra-light airplane.

Chapter 6: *Evolutionary Approach*. One of the major problems faced by engineers willing to use bio-inspiration in the process of hand-crafting artificial systems is the overwhelming amount of details and varieties of biological models. An alternative approach is to apply ideas from the evolutionary process that eventually generated those animals. The so-called artificial evolution embodies this idea of transcribing Darwinian principles into artificial systems. In this chapter, this alternative level of bio-inspiration is used to evolve neuromorphic controllers for vision-based navigation. After preliminary experiments on wheels, the method is then applied to the blimp robot. The evolutionary process does not produce solutions based on optic flow. Nevertheless, efficient obstacle avoidance and handling of critical situations is achieved using the same sensory modalities as in chapter 5, namely vision and gyroscopic information.

Chapter 7: *Concluding Remarks*. The dissertation concludes with a summary of the main results, potential applications and proposal for future work.

Finally, two appendices describe the blimp dynamics modelling (used for flight simulation) and the embedded neural network (used in evolutionary experiments). At the end of this document, an index provides references to important terms and the meaning of the acronyms and abbreviations used in the dissertation.

# Chapter 2

## Biological Principles of Flying Insects

The best model of a cat for biologists is another or better, the same cat.

---

N. Wiener (1894-1964)

### Abstract

This chapter reviews biological principles related to flight control in insects. Looking for biological principles that are amenable to artificial implementation in indoor aircrafts, the review is organised into three topics (or levels of analysis) that are relevant to both robotics and animals: perception (perceptive organs), information processing, and behaviour.

### 2.1 Inspiring Animals

In this thesis, we are mainly interested in flying insects because they face constraints similar to those occurring in miniature aerial robotics, such as minimal power consumption, ultra-low weight, and control of fast motion in real time. Since “flying insects” is a vague notion, we propose to rely on taxonomy to clarify which kind of flying insects are of interest for the synthesis of our artificial systems.

The animal kingdom is divided into phyla, among which are the arthropods composed of four classes, one of those being the insects.

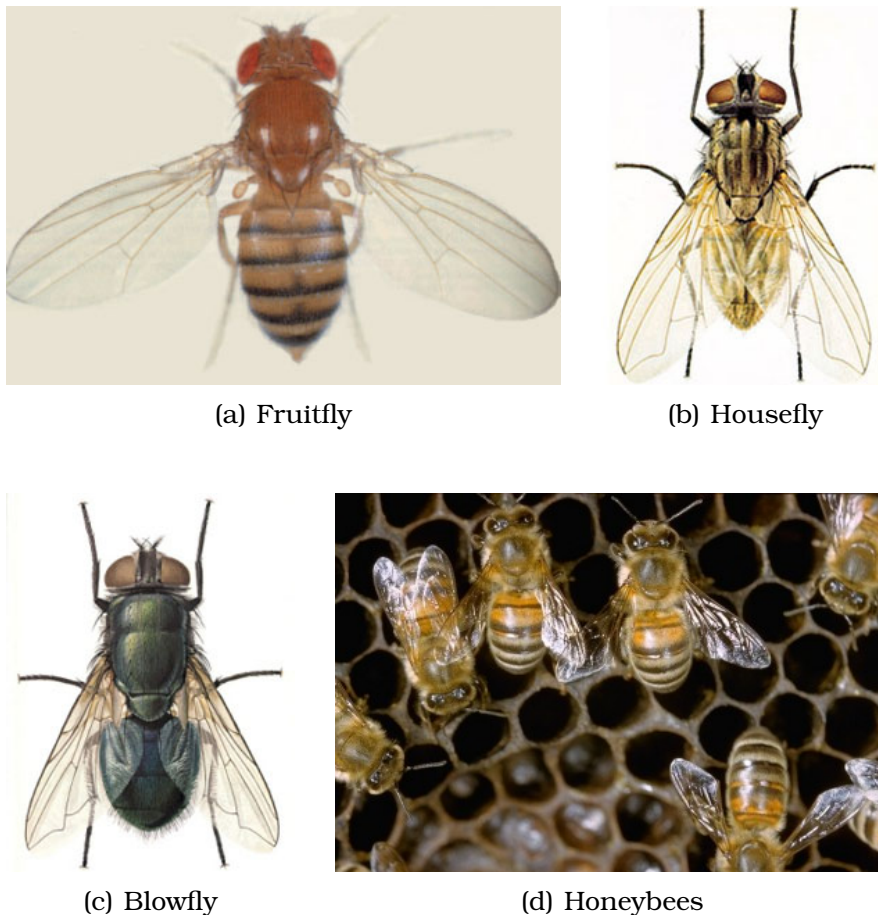


Figure 2.1: A set of highly capable and well-studied flying insects. (a) Fruitfly (*Drosophila*), (b) housefly (*Musca*), (c) blowfly (*Calliphora*), and (d) honeybees (*Apis*).

Arthropods are a phylum of invertebrate animals that have an exoskeleton, a segmented body, and jointed legs. The compound eyes of arthropods are built quite differently from the vertebrate eyes. They are made up of repeated units, the *ommatidia*, each of which functions as a separate visual receptor with its own lens (subsection 2.2.1).

Among arthropods, the most successful flying animals are found in the insect class, which is itself divided into orders such as Diptera (flies and mosquitoes), Hymenoptera (bees), Orthoptera (grasshoppers), Coleoptera (beetles), Lepidoptera (butterflies), Isoptera (termites), Hemiptera (true bugs), etc. In this dissertation, we focus mainly on Diptera and Hymenoptera, not only because flies and bees are generally considered good flyers, but also because a few species of these two orders, namely the blowflies (*Calliphora*), the houseflies (*Musca*),



the fruitflies (*Drosophila*), and the honeybees (*Apis*), have been extensively studied in biology (figure 2.1). Almost all insects have two pairs of wings, whereas Diptera feature only one pair and their hind wings have been transformed through evolution into tiny club-shaped mechanosensors, named *halteres*, which provide gyroscopic information (subsection 2.2.2).

The sensory and nervous systems of flies have been analysed for decades, resulting in a wealth of electrophysiological data, models of information processing and descriptions of behaviours. For example, many neurons in the fly's brain have been linked to specific visually-guided behaviours (Egelhaaf and Borst, 1993a). Although honeybees are capable of solving a great variety of visually controlled tasks (Srinivasan et al., 1996, 2000), comparatively little is known about the underlying neuronal basis. However, interesting models of visually guided strategies are available from behavioural studies.

Perception and action are part of a single closed loop rather than separate entities, but subdividing this loop into three levels helps to highlight the possibilities of artificial implementation. At the first level, anatomical description of flying insects will be a source of inspiration for constructing the robot. Although this thesis is not oriented toward mechanical biomimeticism, the choice of sensor modalities (chapter 3) is based on perceptive organs used by insects. At the second level, models of biological information processing will guide us in the design of sensory signal processing (chapter 4). Mainly related to vision, these models have been essentially produced from neurophysiological studies or from behavioural experiments with tethered animals (see, e.g., Egelhaaf and Borst, 1993a). At the third level, the study of free-flight behaviour (ethology) will give significant insight into how insects steer in their environments and manage to take full advantage of their sensor characteristics by using specific, stereotyped movements. Similar behaviours will be implemented on robots in chapter 5.

In the rest of this chapter, existing descriptions of biological principles are reviewed following the same three levels. However, this brief overview is not an extensive description of flying insect biology. Only models relevant to the simple behaviours described in introduction (attitude control, course stabilisation, obstacle avoidance, and altitude control) and that, we believe, are potentially useful for indoor aircrafts are presented.

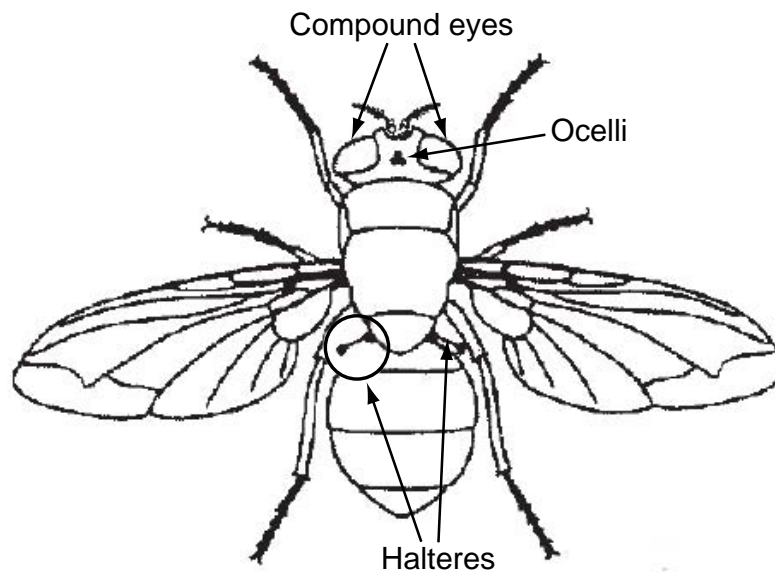


Figure 2.2: The most important perceptive organs related to flight control: the large compound eyes, the ocelli, and the halteres. Outline of the blowfly *Calliphora* adapted from Nalbach (1993).

## 2.2 Sensor Suite for Flight Control

Insects have sense organs that allow them to see, smell, taste, hear and touch their environment (Chapman, 1998). In this section, we focus on the sensors that are known to play an important role in flight control. Whereas flying insects use many sensor modalities, their behaviour is mainly dominated by visual control. They use visual feedback to stabilise flight (Egelhaaf and Borst, 1993b), control flight speed (Srinivasan et al., 1996; Srinivasan and Zhang, 2000), perceive depth (Srinivasan et al., 1991; Tammero and Dickinson, 2002b), track objects (Egelhaaf and Borst, 1993b), land (Borst, 1990; Srinivasan et al., 2000), measure self-motion (Krapp and Hengstenberg, 1996; Krapp, 2000), and estimate distances travelled (Srinivasan et al., 2000). Therefore, the compound eye will be first presented together with the ocelli, a set of three photosensitive organs arranged in a triangle on the dorsal part of the head (figure 2.2). Following that, the gyroscope of Diptera, the halteres, will be described because it is believed to provide important input for course stabilisation. The last subsection of this review is devoted to other perceptive organs that are probably important for flight control but far less understood and more tightly related to aerodynamics rather than navigation control.

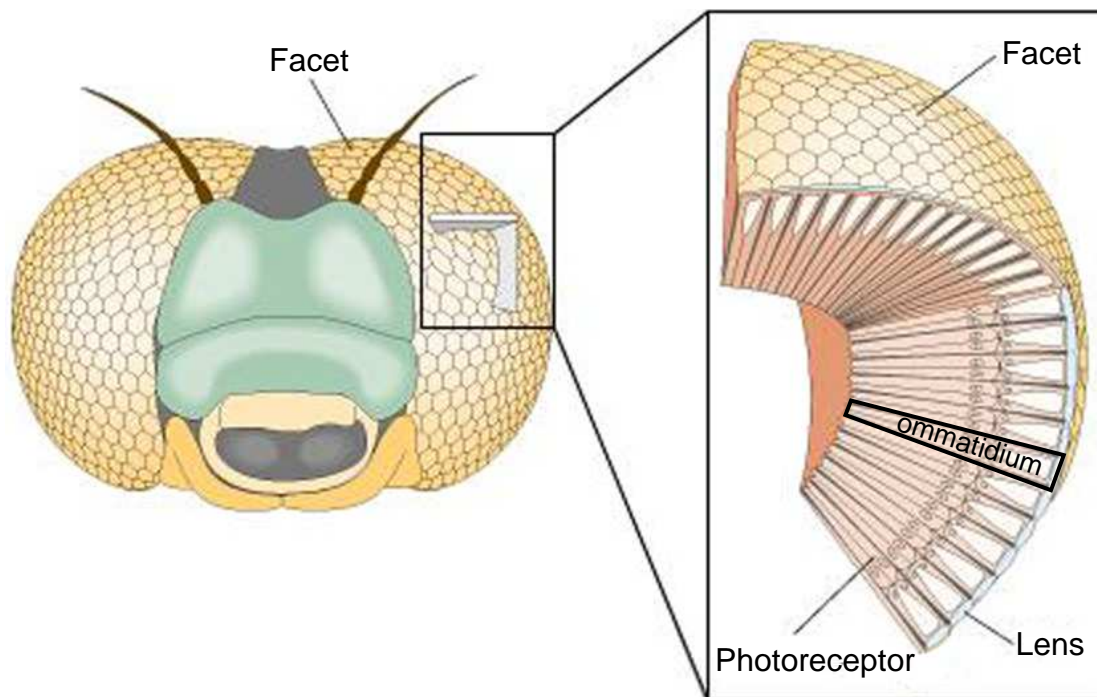


Figure 2.3: The compound eyes of flying insects. The compound eyes are made up of repeating units, the ommatidia, each of which functions as a separate visual receptor. Each ommatidium consists of a lens (the front surface of which makes up a single facet), a transparent crystalline cone, light-sensitive visual cells arranged in a radial pattern, and pigment cells which separate the ommatidium from its neighbours. Adapted from <http://soma.npa.uiuc.edu>.

### 2.2.1 Eyes

Flying insects (and more generally arthropods) have two large compound eyes (Chapman, 1998, p.587) that occupy most of their head (figure 2.2). Each eye is made up of tiny hexagonal lenses, also called facets, that fit together like the cells of a honeycomb (figure 2.3). Each lens admits a small part of the total scene that the insect sees. All the parts together combine and form the whole picture. Underlying the lens is a small tube, the ommatidium, containing several photosensitive cells (for details, see Franceschini, 1975). For the sake of simplicity, in this dissertation we assume that one ommatidium corresponds to one viewing direction and thus to one pixel, although different kinds of compound eyes exist with different arrangements (Land, 1997). In insects, the number of ommatidia varies from about 6 in some worker ants up to 30'000 in some dragonflies. In Diptera, this range is smaller and varies from 700 in the fruitfly to 6000 ommatidia per eye in the blowfly, covering roughly 85% of the visual field (maximum possible

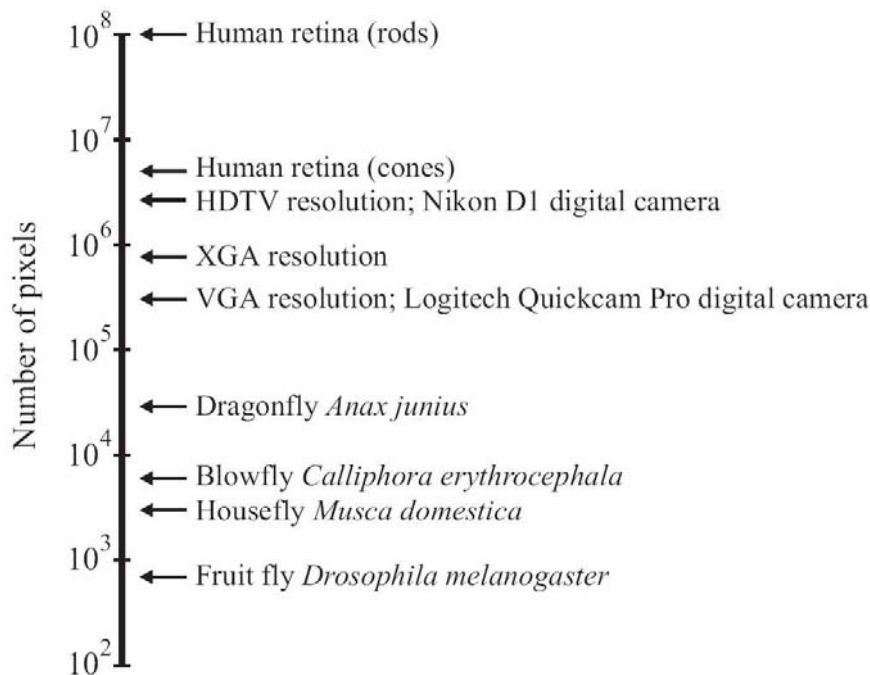


Figure 2.4: Number of pixels in artificial and biological vision systems (single eyes). Flying insects have orders of magnitude fewer pixels than current silicon imagers. Graph from Harrison (2000), insect data from Land (1997).

solid angle whose apex is located at the center of the eye). Taking the square root of the number of ommatidia, the eye of the fruitfly is thus roughly equivalent to a 26x26 pixel array covering one visual hemifield, which is ridiculously low compared to state-of-the-art artificial vision sensors (figure 2.4).

To compare the resolution power of vision systems, one has to consider not only the number of pixels but also the covered field, or more precisely the ratio of the field of view to the number of pixels. According to Land (1997), many flying insects have an interommatidial angle in the range 1-5° (blowfly: 1.1°, housefly: 2.5°, fruitfly: 5°), and this angle corresponds to the visual space a single ommatidia is able to sample (acceptance angle). The best resolving power achievable by the fly's eye, is thus 60 times worst than human eyes, which can easily resolve 1' of arc. However, the compound eye configuration permits a much wider field of view (FOV) because of the juxtaposition of small tubes aimed at different orientation instead of a single lens and a focal plane. Indeed, flies can see in almost any direction except for the blind spot caused by their body.<sup>1</sup>

<sup>1</sup>See Neumann (2002) for a nice reconstruction of fly vision.

It is remarkable that flies are capable of such impressive navigation when considering their low-resolution, which is a consequence of the compound eye design. Moreover, because of their eye arrangement they cannot estimate distances from stereo-vision or focus, as outlined by Srinivasan et al. (1999):

Unlike vertebrates, insects have immobile eyes with fixed-focus optics. Thus, they cannot infer the distance of an object from the extent to which the directions of gaze must converge to view the object, or by monitoring the refractive power that is required to bring the image of the object into focus on the retina. Furthermore, compared with human eyes, the eyes of insects are positioned much closer together, and possess inferior spatial acuity. Therefore the precision with which insects could estimate the range of an object through binocular stereopsis would be much poorer and restricted to relatively small distances, even if they possessed the requisite neural apparatus.

However, fly vision far exceeds human eyes in the temporal domain. Human vision is sensitive to temporal frequencies up to 20Hz, whereas ommatidia respond to temporal frequencies as high as 200-300Hz (Dudley, 2000, p.206). This allows flying insects to be very good at detecting changes in the visual field, which will be refer to as *optic flow* (see section 2.3 and chapter 4).

In addition to their compound eyes, many insects have three simple photoreceptors, called *ocelli*. These ocelli are set in the form of a triangle between the compound eyes (figure 2.2). Since they are unfocused, they cannot form images. Rather, they are used to measure brightness and are thought to contribute to the dorsal light response where the fly aligns its head with sources of brightness (Schuppe and Hengstenberg, 1993). Therefore, ocelli might be used to provide information about location of the horizon in outdoor environments.

### **2.2.2 Halteres**

In many fast moving animals inputs from mechanosensory organs (such as the vertebrate labyrinth) contribute to compensatory reactions, being generally faster than what can be detected through the visual system and independent of lighting conditions (Nalbach and Hengstenberg, 1994). Diptera possess a remarkable organ for measuring an-

gular velocities (Chapman, 1998, p.196). Their body rotations are perceived through the halteres (figure 2.2), which evolved by transformation of the hind wings into tiny club-shaped organs that oscillate during flight in antiphase with the wings (Nalbach, 1993). This mechanosensor measures angular velocity by sensing the periodic Coriolis forces that act upon the oscillating haltere when the fly rotates (Hengstenberg, 1991). Coriolis effects are inertial forces acting on bodies moving in a non-inertial (rotating) reference frame (see also appendix A.5). The forces measured by the halteres are proportional to the angular velocity of the fly's body.

Although the vast majority of insects can fly quite well without halteres, the tiny structures appear necessary for stable flight in Diptera. According to Dickinson (1999), haltere feedback has two roles. The first one is gaze stabilisation:

One important role of the haltere is to stabilize the position of the head during flight by providing feedback to the neck motor system. [...] Nalbach and Hengstenberg demonstrated that the blowfly, *Calliphora erythrocephala*, discriminates among oscillations about the yaw, pitch and roll axes and uses this information to make appropriate compensatory adjustments in head position ([...]; Nalbach, 1993; Nalbach and Hengstenberg, 1994). Such reflexes probably act to minimize retinal slip during flight, thereby stabilising the image of the external world and increasing the accuracy with which the visual system encodes motion.

The second role of halteres consists in direct flight stabilisation:

Although the role of the haltere in stabilising gaze may be important, a more essential and immediate role of the haltere is to provide rapid feedback to wing-steering muscles to stabilize aerodynamic force moments.

In summary, flight stabilisation in flies is ensured by a combination of visual and vestibular senses (section 2.4) and both sensory modalities are of interest for the realisation of artificial systems.

### **2.2.3 Other Mechanosensors**

Although less thoroughly studied, it is likely that flying insects integrate information from other perceptive organs to control flight. One of

those are the campaniform sensilla (Chapman, 1998, p.195) that act as strain gauges. About 335 of them are indeed located at the haltere base for detecting Coriolis forces (Harrison, 2000). Campaniform sensilla are also present on the wings allowing to perceive wing load (Hengstenberg, 1991).

Aerodynamically induced bending in external structures such as antennae potentially provides information about the changing speed and direction of flight (Dudley, 2000). As noted by Hausen and Egelhaaf (1989), antennae are likely to participate in the mechanosensory feedback. Flying insects are also equipped with plenty of tiny bristles that could help in optimising flight by providing information about air movements and changes in air pressure. In their experiment on interaction between vision and haltere feedback, Sherman and Dickinson (2004) remarked:

Posternal hairs on the neck, and wing campaniform sensilla could contribute to both the basic response to mechanical oscillation and the attenuation of the visual reflex during concurrent presentation.

In his thesis, Harrison (2000) also presumes that flies are able to estimate linear acceleration through proprioceptive sensors equipping the legs and neck, which are able to measure position and strain.

Because of the lack of descriptions of how those mechano-sensors interact with flight behaviours, the remainder of this dissertation will primarily focus on vision and on gyroscopic information.

## **2.3 Information Processing**

Among the sensory modalities involved in flight control, visual cues exert the predominant influence on insect orientation and stability. This section will thus focus on vision processing. The importance of vision for flight is underlined by the relative size of the brain region dedicated to processing of afferent optical information (see following paragraph). The fly visual system has been investigated extensively by means of behavioural experiments and by applying neuroanatomical and electrophysiological techniques. Both the behaviour and its underlying neuronal basis can sometimes be studied quantitatively in the same biological system under similar stimulus conditions (Krapp, 2000). Moreover, the neuronal system of flying insects is far simpler

than that of vertebrates, ensuring a better chance to link behaviour to single neuron activity. For instance, the direct neuronal chain between the eye and the flight muscles consists of only 6-7 cells (Hausen and Egelhaaf, 1989). When electrophysiological investigations are not possible because, e.g., of the small size of the neurons, it is still sometimes possible to deduce a mathematical model of the hypothetical functioning of neuronal circuits by recording from higher-level, downstream neurons.

### 2.3.1 Optic Lobes

The optic lobes (i.e., peripheral parts of the nervous system in the head, see figure 2.5) of flies are organised into three aggregates of neurones (also called ganglia or neuropils), namely the *lamina*, the *medulla*, and the *lobula complex* (lobula and lobula plate), corresponding to three centers of vision processing. The retinotopic<sup>2</sup> organisation is maintained through the two first neuropils down to the third one, the lobula, where massive spatial integration occurs and information from very different viewing directions are pooled together.

- The *lamina* lies just under the receptor layer of the eye and receives direct input from photoreceptors. The neurons in this ganglion act as high-pass filters by amplifying temporal changes. They also provide a gain control functionality that ensures quick adaptation to varying background light. Axons from the lamina invert the image from front to back while projecting to the medulla.
- Cells in the *medulla* are extremely small and difficult to record from (see, e.g., Douglass and Strausfeld, 1996). However, behavioural experiments suggest that local motion detection (subsection 2.3.2) occurs at this level. The retinotopic organisation is still present in this second ganglion and there are about 50 neurons per ommatidia. The medulla then sends information to the lobula complex.
- The third optic ganglion, the *lobula complex*, is the locus of massive spatial convergence. Information from several thousand photoreceptors, preprocessed by the two previous ganglia, converges onto a mere 60 cells in the lobula plate (Hausen and Egelhaaf,

---

<sup>2</sup>The neighbourhood is respected, that is, neurons connected to neighbouring ommatidia are next to each other.



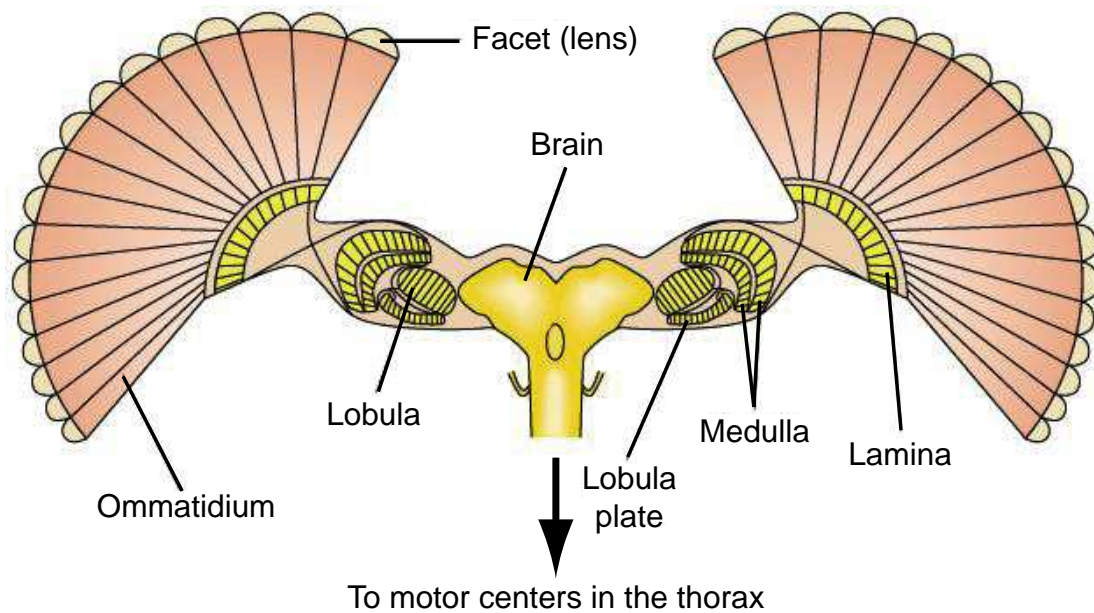


Figure 2.5: The fly's visual and central nervous system (cross section through the fly's brain). Photoreceptors signals are transmitted to the lamina, which accentuates temporal changes. A retinotopic arrangement is maintained through the medulla. The lobula plate is made up of wide-field, motion-sensitive tangential neurons that send information to the contralateral optic lobe as well as to the thoracic ganglia, which control the wings. Adapted from <http://soma.npa.uiuc.edu>.

1989). These so-called *tangential cells* (or LPTC for Lobular Plate Tangential Cells) have broad dendritic trees that receive synaptic inputs from large regions of the medulla, resulting in large visual receptive fields (subsection 2.3.3). The lobula complex projects to higher brain centers and to descending neurons that carry information to motor centers in the thoracic ganglia.

From an engineering perspective, the lamina provides classical functionalities of image preprocessing like temporal and spatial high-pass filtering and adaptation to background light. Although useful in general, those functionalities will not be further described nor implemented in our artificial systems because of the relative visual simplicity of the experimental environments (section 3.4). The two following ganglia, however, are more interesting since they feature typical properties used by flying insects to control their flight. Specificities of medulla and lobula shall thus be further described in the two following paragraphs.

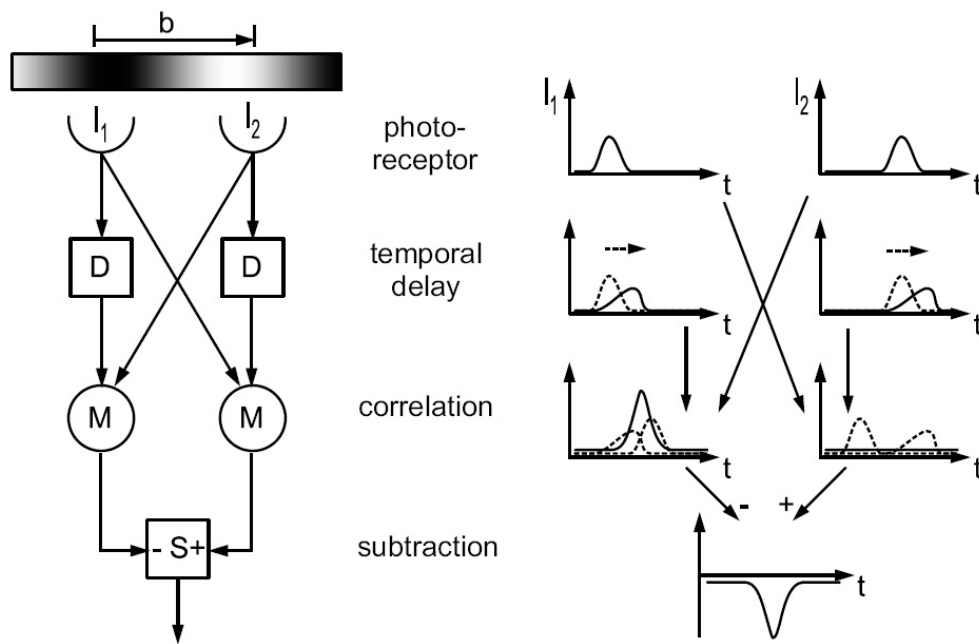


Figure 2.6: Correlation-type elementary motion detector after Reichardt (1969). See text for details. Outline reprinted from Neumann and Bülthoff (2002).

### 2.3.2 Local Motion Detection

Although the use of image motion (or optic flow) in insects is widely recognised as the primary visual cue for in-flight navigation, the neuronal mechanisms underlying local motion detection in the medulla are still elusive (Franceschini et al., 1989; Single et al., 1997). However, behavioural experiments coupled with recordings from the tangential cells in the lobula led to the proposal of functional models of local motion detection. The best-known is the so-called *correlation-type elementary motion detector* (EMD) first proposed by Hassenstein and Reichardt (1956), in which intensity changes in neighboring ommatidia are correlated (Reichardt, 1961, 1969). This model has been initially proposed to account for the experimentally observed *optomotor response* in insects (Götz, 1975). This behaviour tends to stabilise the insect's orientation with respect to the environment and is evoked by the apparent movement of the visual environment.

An EMD of the correlation type basically performs a multiplication of input signals received by two neighbouring photoreceptors (figure 2.6). Prior to entering the multiplication unit, one of the signals is delayed (usually by a first order low-pass filter), whereas the other remains unaltered. Due to these operations, the output of each multiplication unit

responds preferentially to visual stimuli moving in one direction. By connecting two of them with opposite directional selectivities as excitatory and inhibitory elements to an integrating output stage, one obtains a bidirectional EMD (see also Borst and Egelhaaf, 1989, for a good review of the EMD principle). This popular model has been successful at explaining electrophysiological responses of tangential cells to visual stimuli (see, e.g., Egelhaaf and Borst, 1989) and visually-elicited behavioural responses (see, e.g., Borst, 1990).

On the other hand, it is important to stress that this detector is not a pure velocity detector. Indeed, it is sensitive to the contrast frequency of visual stimuli and therefore confounds the angular velocity of patterns with their spatial structure (Reichardt, 1969; Egelhaaf and Borst, 1989; Franceschini et al., 1989; Srinivasan et al., 1999)<sup>3</sup>. The correlation-type EMDs are tuned to temporal frequency, rather than to angular velocity, as outlined by the summary of the optomotor response experiment in figure 2.7.

Although visual motion processing in insects has been studied and characterised primarily through the optomotor response, alternative techniques have led researchers to contradictory conclusions about local motion detection. In the 1980s, stimulating the fly retina at the single photoreceptor level and recording activity of a tangential cell in the lobula, Franceschini and colleagues arrived at a different scheme of local motion detection based on lateral facilitation of a high-pass filtered signal (Franceschini et al., 1989; Franceschini, 2004). The basic idea is that an intensity change detected by a photoreceptor provokes a slowly (exponentially) decaying signal that will be sampled by an impulse due to the same intensity change arriving at the neighbouring photoreceptor. This “facilitate-and-sample” scheme was later implemented in an analog VLSI vision chip (Kramer et al., 1995; Indiveri et al., 1996).

Other studies with free-flying bees have identified several other visually elicited behaviours that cannot be explained by optomotor response and the correlation-type EMD model. These behaviours are the centering response where honeybees tend to balance image velocity on either side when negotiating a narrow gap, the regulation of flight speed, and the landing strategy (see subsection 2.4.4 for further description). They all appear to be mediated by a motion detection mechanism that is sensitive primarily to the speed of the visual stimulus, regardless of its spatial structure or the contrast frequency that it produces (Srini-

---

<sup>3</sup>However, recent work has shown that for natural scenes, enhanced Reichardt EMDs could produce more reliable estimates of image velocity (Dror et al., 2001).

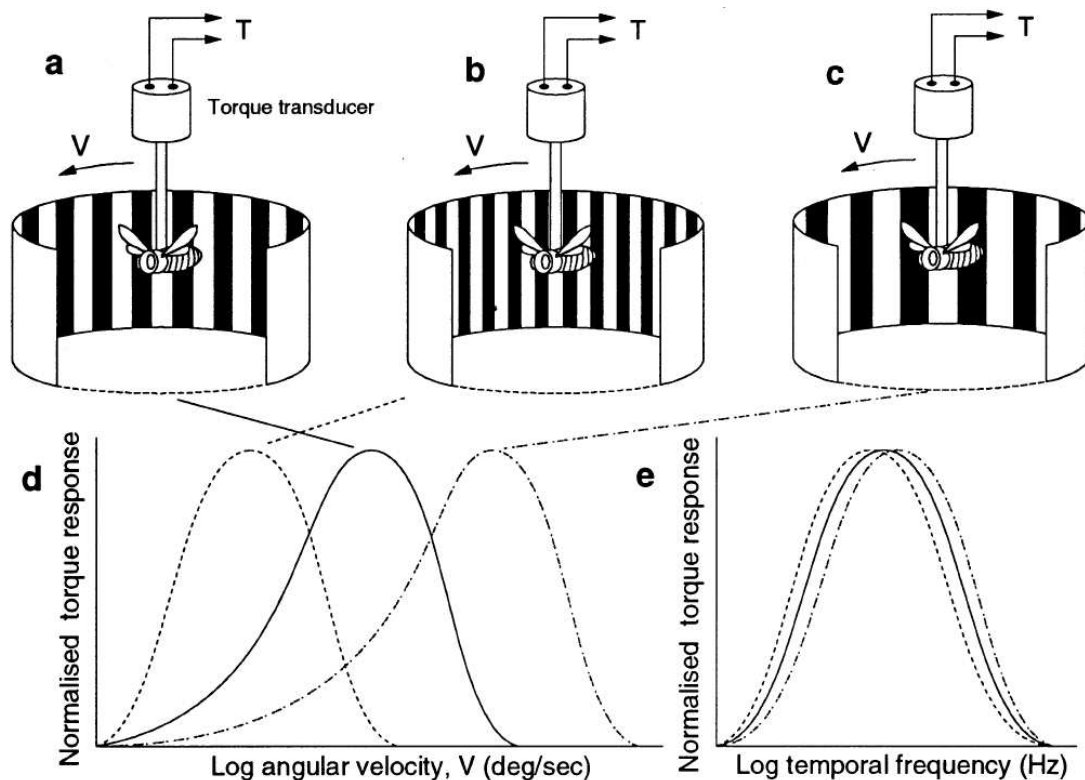


Figure 2.7: The optomotor response of insects. If a flying insect is suspended in a rotating striped drum, it will attempt to turn in the direction of rotation of the drum. The resulting yaw torque is a measure of the strength of the optomotor response. For stripes of a given angular period (as in (a)), the normalised strength of the optomotor response is a bell-shaped function of the rotational speed of the drum, peaking at a specific angular velocity of rotation (solid curve, (d)). If the stripes are made finer (as in (b)), one obtains a similar bell-shaped curve, but with the peak shifted toward a lower angular velocity (dashed curve, (d)). If they are made coarser (as in (c)), the peak response occurs at higher angular velocities (dot-dashed curve, (d)). However, the normalised response curves coincide with each other if they are re-plotted to show the variation of response strength with the temporal frequency of optical stimulation that the moving striped pattern elicits in the photoreceptors, as illustrated in (e). Thus, the optomotor response that is elicited by moving striped patterns is tuned to temporal frequency rather than to angular velocity. Adapted from Srinivasan et al. (1999).

vasan et al., 1999). This finding is supported by an experiment with free-flying *Drosophila* where the flies demonstrated a good insensitivity to spatial frequency when keeping ground speed constant by maintaining image motion at a preferred value, while presented with various upwind intensities (David, 1982).

A neurobiologically realistic scheme for measurement of the angu-

lar speed of an image, independent of its structure or contrast, has been proposed (Srinivasan et al., 1991). This non-directional model is still hypothetical, although recent physiological studies have highlighted the existence of distinct pathways in the optic lobes responsible for directional and non-directional motion detection (Douglass and Strausfeld, 1996). Unlike Reichardt's (correlation-type) and Franceschini's (facilitate-and-sample) models, Srinivasan's model fairly accurately encodes the absolute value of image velocity but not the direction of motion. Note that non-directional motion detection is sufficient, at least for some of the above-mentioned behaviours, like the centering response.

It is interesting to notice that the Reichardt model is so well established that it has been widely used in bio-inspired robotics (e.g., Huber, 1997; Harrison, 2000; Neumann and Bülthoff, 2002; Reiser and Dickinson, 2003; Iida, 2003), although some noticeable departures from it exist (Weber et al., 1997; Franz and Chahl, 2002; Ruffier and Franceschini, 2004). In our case, after (unreported) preliminary trials with artificial implementation of correlation-type EMDs, it became clear that more accurate image velocity detection (i.e., independent of image contrast and spatial frequency) would be needed for the aircrafts. We therefore looked for non-biologically-inspired algorithms producing accurate and directional image motion estimates and selected the image interpolation algorithm (also proposed by Srinivasan, see chapter 4). To clearly notify the difference, instead of using the term EMD, we will use optic flow detector (OFD) to refer to the implemented scheme for local motion detection. Of course, the fact that local motion detection is required as a preprocessing stage in flying insects is widely accepted among biologists and is applied to our bio-inspired robots as well, even if the scheme for computing local optic flow is not proposed in biology.

### **2.3.3 Analysis of Motion Fields**

Visual motion stimuli occur in a stationary environment when the insect is moving because during self-motion the retinal images are continually displaced. The resulting retinal motion patterns depend in a characteristic way on the trajectory described by the insect and on the 3D structure of the visual surroundings. These motion patterns therefore contain information that tells the insect about its own motion or the distances to potential obstacles. However, this information cannot be directly retrieved at the local level and optic flow from different re-

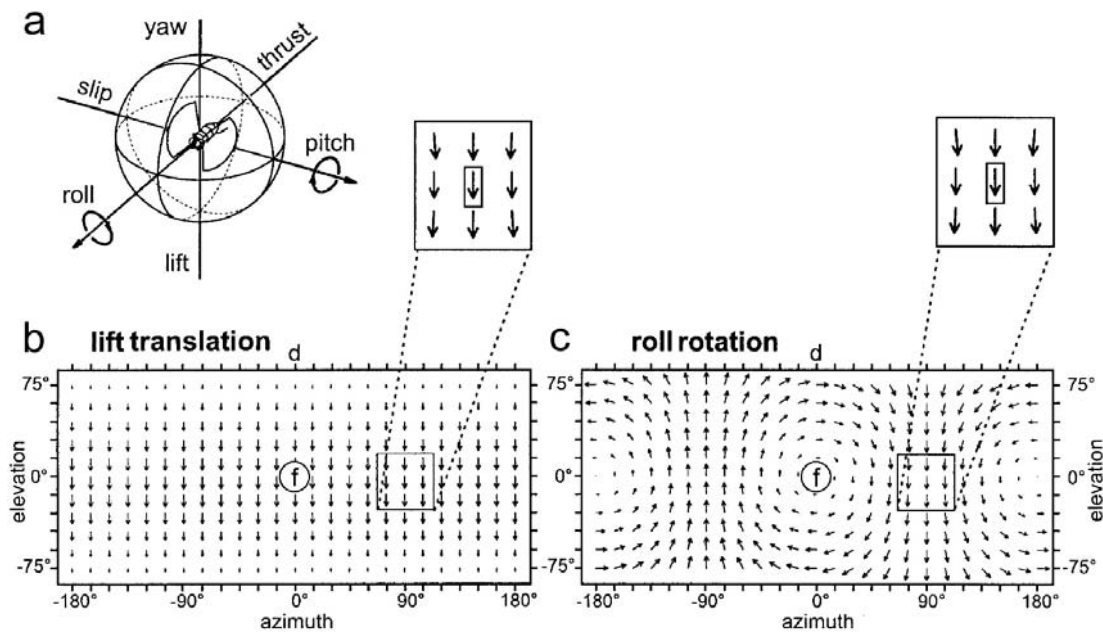


Figure 2.8: Global structure of translatory and rotatory optic flow fields. (a) Motions of a fly can be described by their translatory (thrust, slip, lift) and rotatory (roll, pitch, yaw) components around the 3 body axes (longitudinal, transverse, vertical). The different motion components induce different optic flow fields over both eyes of the moving animal. For simplicity, equal distances from the objects in a structured environment are assumed. (b) Optic flow field caused by a lift translation. (c) Optic flow field caused by a roll rotation. Optic flow patterns are transformed from the visual unit sphere into Mercator maps to show the entire visual space. Visual directions are defined by the angles of azimuth and elevation. The encircled f (frontal) denotes the straight-ahead direction of the fly. Globally, the two optic flow fields can easily be distinguished from one another. However, this distinction is not possible at the level of local motion detectors. See, e.g., the optic flow vectors indicated in the boxes: local motion detectors at this place would elicit exactly the same response irrespective of the motion. Adapted from Krapp et al. (1998).

gions of the visual field must generally be combined in order to infer behaviourally significant information (see, e.g., figure 2.8).

Analysis of the global motion field (or at least several different regions) is thus generally required in order for the local measurements to be exploited at the behavioural level. Some sort of spatial integration is known to happen after the medulla (where local motion detection occurs retinotopically), mainly in the lobula plate where tangential neurons receive input from large receptive fields (Hausen and Egelhaaf, 1989). The lobula plate thus represents a major centre of optic flow field analysis. Some of the 60 neurons of the lobula plate are

known to be sensitive to coherent large-field motion (i.e., the VS, HS and Hx-cells), whereas other neurons, the figure detection cells (FD-cells), are sensitive to relative motion between small objects and background (Egelhaaf and Borst, 1993b; Krapp and Hengstenberg, 1996). As an example of the usefulness of these neurons at the behavioural level, there is good evidence that HS and VS-cells are part of the system that compensates for unintended turns of the fly from its course (Krapp, 2000).

### **Detection of Self-motion**

Quite recently, neuroscientists have analysed the specific organisation of the receptive fields, i.e., the distribution of local preferred directions and local motion sensitivities, of about 30 tangential cells in the lobula. They found that the response fields of VS neurons are not uniform but resemble rotatory optic flow fields that would be induced by the fly during rotations around various horizontal axes (Krapp et al., 1998). In contrast to the global rotatory structure within VS response fields, the response field of Hx cells shows the global structure of a translatory optic flow field (Krapp and Hengstenberg, 1996). The response fields of HS cells are somewhat more difficult to interpret since they probably do not discriminate between rotatory and translatory components (Krapp, 2000). In summary, all is happening as if tangential cells were acting as neuronal matched filters (Wehner, 1987) tuned to particular types of wide-field motion (figure 2.9). It is also interesting to notice that those receptive-field organisations are highly reliable at the interindividual level (Krapp et al., 1998) and seem to be independent of early sensory experience of the fly. That means that the specificity of these cells to optic flow fields have probably evolved on a phylogenetic time scale (Karmeier et al., 2001).

Franz and Krapp (2000) had some success at estimating self-motion of a simulated agent based on this theory of visual matched filters. However, Krapp (2000) warns us about too simplistic interpretations of this biological model of spatial integration:

[Some] approaches take for granted that the results of the local motion estimates are summed up in a linear fashion at an integrating processing stage. For insect visual systems, however, it was found that local motion analysis is achieved by elementary motion detectors whose output is not simply proportional to velocity [...] but also depends on pattern

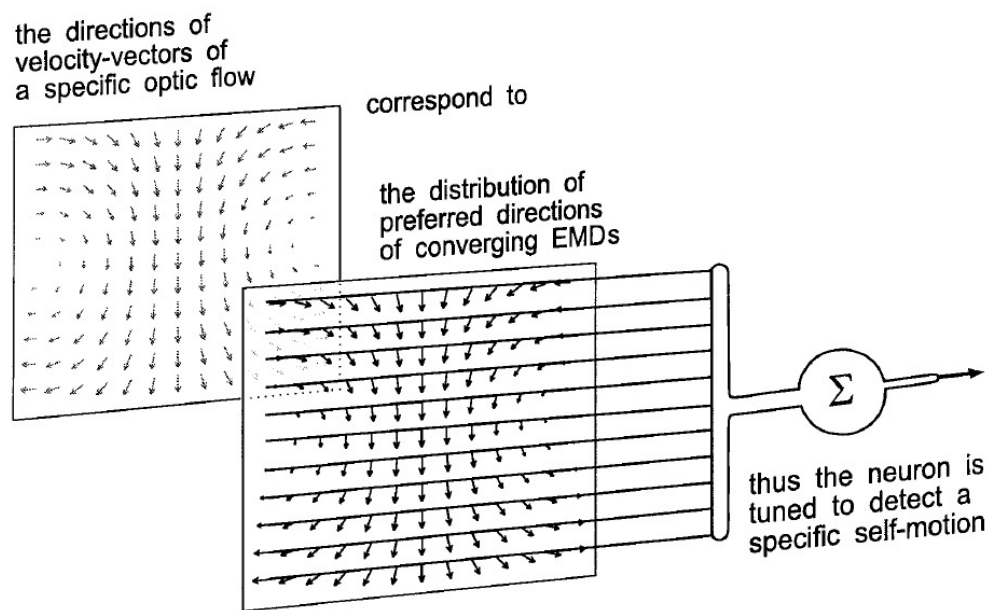


Figure 2.9: Hypothetical filter neuron matched to a particular optic flow field induced by self-motion (e.g., rotation). Local motions of the optic flow field locally activate those motion detectors with appropriate preferred directions. A wide-field neuron selectively collects and spatially integrates the signals of these motion detectors. Hence it would be most sensitive to that particular optic flow and consequently to the self-motion that caused the flow. Adapted from Krapp et al. (1998).

properties like spatial wavelength and contrast [...]. Hence, it remains unclear how biological sensory systems cope with highly dynamic stimuli as encountered, for instance, by the fly during free flight. It is by no means easy to predict the signals of the tangential neurons under such natural conditions.

Another problem is that tangential neurons like the VS cells cannot be expected to be insensitive to optic flow components induced by movements that are not their own preferred self-motion. Output of those neurons needs to be corrected for apparent rotations, which may be due to translatory self motions and to rotations around axes other than the preferred axis. In fact, the use of visual or gyroscopic information for correcting such errors is a recurrent question which is not totally resolved yet. After Krapp (2000),

The signals necessary to correct for these erroneous response contributions could be supplied by other wide field neurons.



Or, alternatively:

Correction signals encoding fast self-rotations may also be supplied by the haltere system (Nalbach, 1994). Because the dynamic range of the haltere system is shifted toward higher angular velocities, it is thought to complement the visual self-motion estimation (Hengstenberg, 1991).

The computational properties of tangential neurons have mainly been characterised in tethered flies with simplistic visual stimuli (e.g., Krapp et al., 1998). A recent study where blowflies were presented with behaviourally relevant visual inputs suggests that responses of tangential cells are very complex and hard to predict based on the results obtained with simplistic stimuli (Lindemann et al., 2003). As explained by Egelhaaf and Kern (2002), only few experiments with natural stimuli have been done and even less in closed-loop situation:

Neuronal responses to complex optic flow as experienced during unrestrained locomotion can be understood only partly in terms of the concepts that were established on the basis of experiments done with conventional motion stimuli. [...] It is difficult to predict the performance of the system during complex flight manoeuvres, even when wiring diagrams and responses to simplified optic-flow stimuli are well established.

### **Perception of Approaching Objects**

Apart from the widely covered topic of tangential cells in the lobula plate and their resemblance to matched filters, another model of wide field integration has been proposed to explain the detection of imminent collision. Here the purpose is about estimating distance from objects or *time to contact* (TTC), rather than detecting self motion. Looming stimuli (expanding images) have long been thought to act as an essential visual cue for detecting imminent collisions (see, e.g., Lee, 1976). When tethered flying flies encounter a looming stimulus, they extend their forelegs in preparation to land. This *landing response* has been shown to be triggered by visual looming cues (Borst and Bahde, 1988). Experiments show that the latency of the landing response is reciprocally dependent on the spatial frequency content and on the contrast of the pattern, as well as the duration of its expansion. Therefore,

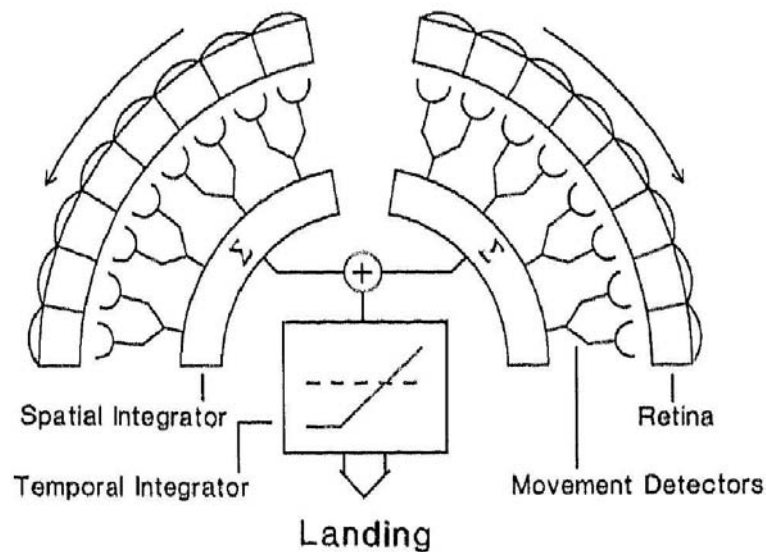


Figure 2.10: The so-called STIM (spatio-temporal integration of motion) model underlying the landing response of the fly. The output of directionally selective correlation-type movement detectors are pooled from each eye. These large-field units feed into a temporal leaky integrator. Whenever the integrated signal reaches a fixed threshold landing is released and a preprogrammed leg motor sequence is performed to avoid crash-landing. Figure from Borst and Bahde (1988).

Borst and colleagues proposed a model based on a spatial integration of correlation-type EMDs (figure 2.10), which present the same kind of dependence on spatial frequency and contrast (subsection 2.3.2). Very recently, Tammero and Dickinson (2002a) have shown that *collision avoidance* manoeuvres could also be explained by the perception of image expansion as detected by an array of local motion detectors (subsection 2.4.3).

So far, neurons extracting image expansion from the retinotopic array of local motion detectors have not been found at the level of the lobula complex (Egelhaaf and Borst, 1993b). In the cervical connective (just below the brain in figure 2.5), however, cells are known to be sensitive to retinal image expansion. These neurons, which respond strongest when the animal approaches an obstacle or a potential landing site, have been proposed to be part of the neuronal circuit initiating the landing response (Borst, 1990).

Other biologists have proposed similar schemes, although based on pure time to contact and thus without any dependency on contrast or spatial frequency, for explaining deceleration of flies before landing (Wagner, 1982) or stretching of wings in plunging gannets (Lee and

Reddish, 1981). From a functional point of view, it would obviously be advantageous to use a strategy that estimates time to contact independently of the spatial structure of the object being approached. Indeed, if the underlying local optic flow detection is a true image velocity detection, the measure of time to contact can be directly extracted from optic flow measurements (Poggio et al., 1991; Ancona and Poggio, 1993; Camus, 1995).

In summary, individual cells (either in the lobula or in the cervical connective) receive inputs from many local motion detectors and generate output signals that appear tuned to estimate particular features of the global optic flow field that flying insects experience during flight. Spatial integration of local optic flow vectors is thus a necessary operation to provide useful information for several behaviours like course stabilisation, landing, obstacle avoidance, etc. In our artificial systems, although the weight limitations of our aircrafts do not permit the presence of many local motion detectors as in flying insects, some kind of spatial integration is used to detect typical patterns of optic flow. In particular, the information processing algorithm for detection of imminent collision of the airplane is based on the STIM model (figure 2.10).

## 2.4 In-Flight Behaviours

As seen before, visual motion and mechanosensors are exploited by insects to gain information on the 3D layout of the environment and the rate of self-motion in order to control behaviours. In this section, a set of four behaviours are reviewed together with links to possible underlying information processing strategies presented in the previous section. This restricted palette of behaviours is not a representative sample of the biological literature, but was selected as a minimal set of control mechanisms that could allow an aircraft to fly in enclosed environments.

### 2.4.1 Attitude Control (ATC)

One of the primary requirements for a flying system is to be able to control its *attitude* in order to stay the right way up or bank to turn (Horridge, 1997). The attitude of an aerial system is defined by its pitch and roll angles (see figure 2.8a). The so-called *passive stability* encompasses simple mechanisms providing flight equilibrium. For

instance, the fact that insect wings are inserted above the center of gravity provides some degree of passive stability around the roll axis (Chapman, 1998, p.214). Other aerodynamic characteristics of the insect body provide partial compensation for unintended pitch torques (Dudley, 2000, p.203). However, in small flapping-wing insects relying on unsteady-state aerodynamics<sup>4</sup>, such passive mechanisms can compensate only for a small subset of unintentional rotations.

Other mechanisms of *active stability* are thus required in insects for attitude control. One of those is the so-called *dorsal light response* (Schuppe and Hengstenberg, 1993) by which insects attempt to balance the level of light received in each of their three ocelli (subsection 2.2.1). This response is believed to help insects at keeping their attitude aligned with the horizon (Dudley, 2000, p.212). Such mechanisms have been proposed for attitude control in simulated flying agents (Neumann and Bülthoff, 2002). For indoor environments, however, we believe that this approach is nearly impossible. If insects controlled their attitude exclusively by means of a dorsal light response, they would sometimes have a tendency to fly at unusual angles when flying among obstacles that partially occlude light sources. The fact that this does not occur indicates that other stimuli are also important, although not yet fully understood (Chapman, 1998, p.216).

It is probable that optic flow (subsection 2.3.3) provides efficient cues for pitch and roll stabilisation in a way functionally similar to the optomotor response (primarily studied for rotations around the yaw axis). However, optic flow depends on angular rate and not on absolute angles. Therefore, such mechanisms fail to provide a reliable vertical reference. The same holds for the halteres (subsection 2.2.2), which are also known to help at regulating pitch and roll velocities but are not able to provide an absolute reference.

In artificial systems, like aircrafts relying on steady-state aerodynamics, passive stabilisation mechanisms are often sufficient to give strong compensation torques to eliminate pitch and roll. For instance, a non-zero angle between the left and right wings (called dihedral, see subsection 3.1.3 for further details) help for maintaining the wings horizontal, whereas a low center of gravity and/or a well-studied tail geometry provides good pitch stability<sup>5</sup>. The aircrafts described in this thesis

---

<sup>4</sup>Direction, geometry and velocity of airflow change over short time intervals.

<sup>5</sup>Note however, that rotorcrafts are far less passively stable than planes and active attitude control of such systems is a delicate issue because proprioceptive sensors like inclinometers are perturbed by accelerations.

operate within the range of steady-state aerodynamics and therefore do not need active attitude control, such as the dorsal light response.

### 2.4.2 Course (and Gaze) Stabilisation (CS)

Maintenance of stable flight trajectory is not only useful for travelling from a point to another, but also to facilitate depth perception of flying insects, as pointed out by Krapp (2000):

Rotatory self-motion components are inevitable consequences of locomotion. The resulting optic-flow component, however, does not contain any information about the 3D layout of the environment. This information is only present within translatory optic-flow fields. Thus for all kinds of long-range and short-range distance estimation tasks, a pure translatory optic flow field is desirable (Srinivasan et al., 1996; [...]). One possibility to, at least, reduce the rotatory component in the optic flow is to compensate for it by means of stabilising head movements and steering maneuvers. These measures can be observed in the fly but also in other visually oriented animals, including humans.

The well-known optomotor response (introduced in subsection 2.3.2), which is evoked by the apparent movement of the visual environment, tends to minimize image rotation during flight and helps the insect to maintain a straight course (Srinivasan et al., 1999). Hence, course stabilisation of flying insects relies essentially on the evaluation of the retinal motion patterns perceived during flight and reviewed in subsection 2.3.3. Haltere feedback is also known to play an important role in course stabilisation as well as in gaze or head<sup>6</sup> orientation. As suggested in Krapp's statement, rapid head compensation helps at cancelling rotational optic flow before the rest of the body has time to react (Hengstenberg, 1991). In the free-flying blowfly, the angular velocities of the head are approximately half those of the thorax during straight flight (van Hateren and Schilstra, 1999) .

The integration of visual and gyroscopic modalities for course and gaze stabilisation is very complicated and not yet fully understood. Chan et al. (1998) have shown that motoneurons innervating muscles of the haltere receive strong excitatory input from visual interneurons

---

<sup>6</sup>In this context, gaze or head control have the same meaning because insect eyes are mostly solidly attached to the head.

such that visually guided flight manoeuvres may be mediated in part by efferent modulation of hard-wired equilibrium reflexes. Sherman and Dickinson (2004) have proposed a model of stabilisation where sensory inputs from halteres and the visual system are combined in a weighted sum. What is better understood is that fast rotations are predominantly detected and controlled by mechanosensory systems whereas slow drift and steady misalignments are perceived visually (Hengstenberg, 1991).

Whatever the sensory modality used to implement it, course stabilisation clearly is an important mechanism in flying systems. On the one hand, it allows to counteract unwanted deviations due to turbulences. On the other hand, it provides the visual system with less intricate optic flow fields (i.e., exempt from rotational components), hence facilitating depth perception.

### 2.4.3 Obstacle Avoidance (OA)

As seen in subsection 2.3.3, an approach trajectory that is almost perpendicular to a surface would generate strong looming cues, which can serve as imminent collision warnings. Work by Wagner (1982), as well as Borst and Bahde (1988), has shown that deceleration and extension of the legs in preparation for landing are triggered by large-field, movement-detecting mechanisms that sense expansion of the image. Instead of landing, flying insects could decide to turn away from the looming object in order to avoid it.

This indeed has been recently studied by Tammero and Dickinson (2002b). The flight trajectories of many fly species consist of straight flight sequences<sup>7</sup> interspersed with rapid changes in heading called *saccades* (Collett and Land, 1975; Wagner, 1986; Schilstra and van Hateren, 1999). Tammero and Dickinson (2002b) have reconstructed the optic flow seen by free-flying *Drosophila*. Based on the results, they proposed a model of saccade initiation using the detection of visual expansion, an hypothesis that is consistent with open-loop presentation of expanding stimuli to tethered flies (Borst, 1990). Although differences in the latency of the collision-avoidance reaction with respect to the landing response suggest that the two behaviours are mediated by separate neuronal pathways (Tammero and Dickinson, 2002a), the STIM model proposed by Borst (1990) and reprinted in figure 2.10 rep-

---

<sup>7</sup>During which the course stabilisation mechanisms described above are probably acting.

resents a good understanding of the underlying mechanism. Several implementations of artificial systems capable of avoiding collisions have been done using a variant of this model. The implementation being most closely inspired by the experiments from Tammero and Dickinson (2002b) was developed in that same laboratory (Reiser and Dickinson, 2003).

#### **2.4.4 Altitude Control (ALC)**

Altitude control in insects is a behaviour that has almost never been directly studied in insects. It represents however an important basic mechanism for roboticists interested in building autonomous flying machines. In this section, we shall thus consider related behaviours that help to understand how an aerial system could regulate altitude using visual motion cues. Those behaviours, particularly studied in honeybees, are the centering response, the regulation of flight speed, and the grazing landing.

Bees flying through a narrow gap or a tunnel have been shown to maintain equidistance to the flanking walls (centering response) by balancing the apparent speeds of the retinal images on either side (Srinivasan et al., 1996, 1997). The experiments reported by Srinivasan et al. (1991) demonstrate unequivocally that flying bees estimate lateral distances from surfaces in terms of the apparent motion of their images, moreover largely independently of their spatial frequency or contrast. In the flight speed regulation behaviour (Srinivasan et al., 1996, 1997; Srinivasan, 2000), the speed of flying bees is shown to be controlled by maintaining constant average image velocity as seen by the lateral regions of the two eyes. This arguably avoids potential collisions by ensuring that the insect slows down when flying through narrow passages. The grazing landing (as opposed to the landing response described in subsection 2.4.3) describes how bees execute a smooth touchdown on horizontal surfaces (Srinivasan et al., 1997, 2000). In this situation, looming cues are weak because the landing surface is almost parallel to the flight direction. Once again, bees are shown to hold the image velocity of the surface in the ventral part of their eyes constant as they approach it, thus automatically ensuring that the flight speed is close to zero at touchdown. These three behaviours clearly demonstrate the ability of flying insects to regulate self-motion using translational optic flow. The advantage of such strategies is that the control is achieved by a very simple process, and without explicit

knowledge of the distance from the surface (Srinivasan and Zhang, 2000).

Observation of migrating locusts have shown that these animals tend to hold constant the optic flow experienced in the ventral part of their eyes (Kennedy, 1951). This ventral optic flow is proportional to the ratio between forward speed and altitude. Taking inspiration from these observations, Ruffier and Franceschini (2003) proposed an altitude control system, an optic flow regulator, that keeps the ventral optic flow at a reference value. At a given ground speed, holding the ventral optic flow constant leads to level flight at a given height. If the forward speed happens to decrease (deliberately or as a consequence of wind), the optic flow regulator produces a decrease in altitude. This optic flow regulator was implemented on a tethered helicopter and demonstrated efficient altitude control and terrain following. Ruffier and Franceschini (2004) also showed that the same strategy could generate automatic takeoff and landing, and suitable descent or ascent in the presence of wind, as actually observed in migrating locusts (Kennedy, 1951).

One of the major problems of such strategies lies, once again, in the perturbation of the translational flow field by rotational components. In particular, every attitude correction will result in rotation around the pitch or roll axes and indeed create rotational optic flow. Hence a system correcting for these spurious signals is absolutely required. In flying insects, this could be accomplished by gaze stabilisation (described in subsection 2.4.2). In artificial systems, the vision system could be actuated and regulated such as to remain vertical (this solution was adopted in Ruffier and Franceschini, 2004). However, such a mechanism requires a way to measure attitude angles in a non-inertial frame, which is a non-trivial task. Another solution consists in measuring angular rates with an inertial system (gyroscope) and directly subtracting rotational components from the global optic flow field.

## 2.5 Summary

Attitude control (subsection 2.4.1) in insects is probably required in order to provide stable reference for using vision during motion (Horridge, 1997); and in turn, vision seems to be the primary cue for controlling attitude. The same holds for course stabilisation (subsection 2.4.2) whereby straight trajectories allow for cancellation of rotatory



optic flow and easier interpretation of optic flow for distance estimation. This shows once again that perception, information processing, and behaviour are tightly connected and organised into a loop where adequate behaviour is not only needed for navigation (and, more generally, survival), but also a prerequisite to efficient perception and information processing. This idea is equally highlighted by biologists like Egelhaaf et al. (2002):

Evolution has shaped the fly nervous system to solve efficiently and parsimoniously those computational tasks that are relevant to the survival of the species. In this way animals with even tiny brains are often capable of performing extraordinarily well in specific behavioural contexts.

Therefore, when taking inspiration from biology, it is worth to look at those different levels as being tightly connected to each other, rather than trying to design artificial systems behaving like animals while featuring highly precise, Cartesian sensors, or, on the other hand, creating robots with biomorphic sensors for cognitive tasks. Following this advice, our robot design takes inspiration from flying insects at the following three levels:

- *Perception.* The choice of sensor modalities is largely based on those of flying insects. Only low-resolution vision and gyroscopic information (chapter 3) are used to feed the control system.
- *Information processing.* In the experiments of chapter 5, the way information is processed is largely inspired from what has been described in this chapter. Visual input is first preprocessed with an algorithm producing local motion detection (chapter 4). Data from local motion estimates is then spatially integrated and combined with gyroscopic information in order to provide the control system with meaningful information.
- *Behaviour.* Based on this preprocessed information, the control system is then designed so to reproduce the insect behaviours presented in section 2.4 that fit exactly that choice of sensors and processing. The resulting system provides robots with basic navigational capabilities (i.e., course stabilisation, obstacle avoidance and altitude control), which enable them to move around autonomously, without colliding with obstacles.



# Chapter 3

## Robotic Platforms and Tools

Flies are objectionable in many ways, but they now add insult to injury by showing that it is definitely possible to achieve the smartest sensory-motor behavior such as 3D navigation at 500 body-lengths per second using quite modest processing resources.

---

N. Franceschini (2004)

### Abstract

This chapter presents a set of three mobile robots developed within this thesis to serve as a basis for assessment of the proposed biologically inspired strategies. They are a miniature wheeled robot for preliminary tests, an indoor airship, and an ultra-light fixed-wing airplane. In spite of the fundamental differences regarding their body shapes, actuators and dynamics, the three robotic platforms share several electronic components in order to ease the transfer of software, processing schemes and control systems from one to another. Obviously, these robots do not attempt to reproduce the bio-mechanical principles of insect flight. However, the perceptive modalities present in flying insect are taken into account in the selection of sensors. Next to the platforms, we also shortly present the software tools used for interfacing the robots and simulating them. We conclude this chapter with an overview of the experimental arenas and their respective characteristics.

## 3.1 Platforms

The three platforms are introduced in order of increasing complexity, and not in the order they will be used for experiments in the following chapters. This section focuses on the structure and dynamic properties of the different robots, whereas the next section presents the electronic components and sensors, which are largely compatible among the three platforms. The minimum set of control mechanisms (according to section 2.4) required by each platform is also mentioned along with their descriptions. At the end of the section, a comparative summary the platforms' main characteristics is provided.

### 3.1.1 Miniature Wheeled Robot

The popular *Khepera* (Mondada et al., 1993) is our “battle horse” for preliminary testing of control strategies. The Khepera is a simple and robust differential-drive robot that has proved to be suitable for long-lasting experiments typical in evolutionary robotics (Nolfi and Floreano, 2000). It can withstand collisions with obstacles, does not overheat when its motors are blocked, and can easily be externally powered

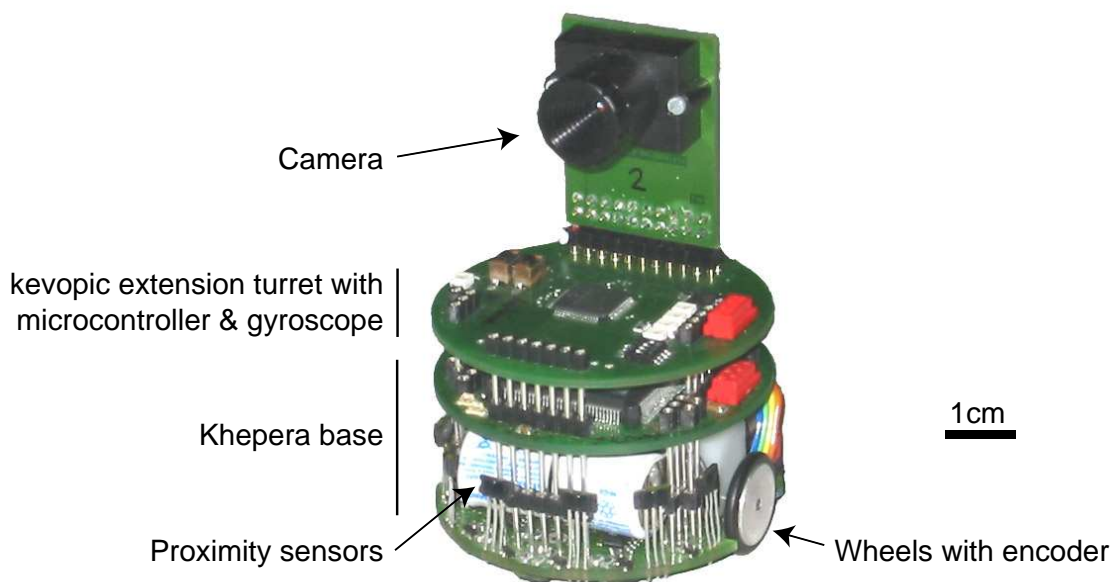


Figure 3.1: The Khepera robot equipped with the custom extension turret *kevopic*.

via a rotating contact hanging above the experimental arena, therefore releasing the experimenter from the burden of changing batteries periodically.

To enable good compatibility with the aerial platforms, the Khepera has been augmented with a custom turret (figure 3.1). The so-called *kevopic* (Khepera, evolution, PIC) turret features the same core microcontroller and interfacing capabilities as the boards equipping the flying robots. Kevopic also supports the same vision and gyroscopic sensors (subsection 3.2.2). This turret has been developed because the basic Khepera microcontroller, a Motorola™ MC68331 was not suitable for flying robots (too heavy and power consuming) and could not be programmed with the same development environment as the microcontroller selected for the aerial platforms (subsection 3.2.1).

The sensing capabilities of the underlying standard Khepera can still be accessed from the kevopic. Beside the two main sensor modalities (vision and gyroscope), this wheeled platform features wheel encoders and 8 infrared proximity sensors. In this thesis, those additional sensors are never used as inputs to the navigation controller because they are neither available in the flying robots nor present in flying insects. However, they are useful in the process of analysing the performances of the bio-inspired controllers. For instance, the proximity sensors allow to know whether the robot is close to arena borders and the wheel encoders enable the plotting of the produced trajectories with a reasonable precision over a relatively short period of time.

The Khepera moves on a flat surface and has 3 degrees of freedom (DOF). Therefore, it is an ideal candidate for testing obstacle avoidance (OA) algorithms without requiring course stabilisation (CS). Since it is in contact with the floor and has negligible inertial forces, the trajectory is determined solely by wheel speeds. It is sufficient to issue the same motor command on the left and on the right wheels in order to move straight. Of course, attitude and altitude control (ATC and ALC) are not required on this robot since it moves on a flat surface. However, in chapter 5, we shall employ the Khepera to demonstrate vision-based ALC by orienting the camera laterally (figure 3.9) and doing wall following. From a bird-eye perspective, the wall replaces the ground and, at first approximation, the heading direction of the Khepera as the pitch angle of an airplane.

### 3.1.2 Indoor Airship

When it comes to flying robots, one has to choose a lift methods among the existing ones: aerostat, fixed-wing, flapping-wing, rotorcraft, and jet-based. The most simplest from a mechanical and structural perspective is probably the aerostat principle.

#### **Blimps as Robotic Platforms**

After Archimedes (see also section A.1 in appendix), a volume surrounded by a fluid (in our case, the ambient air) generates a buoyant force that is equal to the mass of the fluid displaced by this volume. In order to fly, airships are thus lighter than the mass of air occupied by their hull. They achieve that by filling the big volume of their hull with a gas that is far lighter than air (helium is often employed) in order to compensate for the weight of the gondola and equipment that are hanging below the hull. Such a lift principle presents several advantages:

- No specific skills in aerodynamics are needed for building a system that flies. Inflating a bag with helium and releasing it into the air with some balancing weight will produce a minimalist flying platform that just stays airborne like a submarine floats in the water.
- Unlike helicopters or jet-based systems, it is not dangerous for indoor use and far more silent.
- Unlike all other flying schemes, it does not consume energy to stay aloft.
- The envelope size can easily be adapted to the required payload (e.g., a typical spherical Mylar bag of 1m in diameter filled with helium can approximately lift 150g of payload in addition to its own weight).
- An airship is largely stable by nature. Its center of gravity lies below the center of buoyancy, creating restoring forces that keep the airship upright (see subsection A.1 in appendix). If used under reasonable accelerations, an airship can thus be approximated by a 4 DOF platform because pitch and roll are always close to zero degrees.

- Equipped with some protections, a blimp can bump into obstacles without damage while remaining airborne, which is definitely not that trivial for airplanes or helicopters.

All these advantages led several research teams to adopt such a kind of lighter-than-air platform in different areas of indoor robotic control like visual servoing (Zhang and Ostrowski, 1998; van der Zwaan et al., 2002; da Silva Metelo and Garcia Campos, 2003), collective intelligence (Melhuish and Welsby, 2002), or bio-inspired navigation (Planta et al., 2002; Iida, 2003). The same advantages allowed us to setup the first evolutionary experiment entirely performed on a physical flying robot (Zufferey et al., 2002). Note that the version used at that time, the so-called *Blimp1*, was slightly different than the one presented below.

Apart from the need for helium refill from time to time, the main drawbacks of a blimp-like platform reside in its inertia caused by its considerable volume. Because of its shape and dynamics, a blimp is also more different from flying insects than an airplane. This platform was mainly built as an intermediate step between the miniature wheeled robot and the ultra-light winged aircraft to enable aerial experiments that were not possible with the airplane (chapter 6). A blimp already features much more complex dynamics than a Khepera. In particular, it has inertia effects, must counteract air drag and side slip (see appendix A for details).

### **The Blimp2b**

The latest prototype, the so-called *Blimp2b* (figure 3.2), has a helium-filled envelope with a lift capacity of 100g. The almost ellipsoid hull measures 110x60x60cm. The underneath gondola is made of thin carbon rods. Attached to the gondola frame are three thrusters (8mm DC motors, gear and propellers from Didel<sup>TM</sup>) and two sensors in addition to camera and gyroscope that will be described in section 3.2 below. An anemometer made of a free rotating propeller mounted on a shaft with an optical encoder (developed by Didel<sup>TM</sup>) allows to measure forward air-speed. As in the case of the Khepera's encoders, the anemometer is only used for performance estimation and not for navigation control. A vertical distance sensor (Sharp<sup>TM</sup> GP2Y0A02YK) provides an estimate of altitude over ground. On-board energy supply is ensured by a 1200mAh Lithium-polymer battery, which is sufficient for 2-3 hours of autonomy.

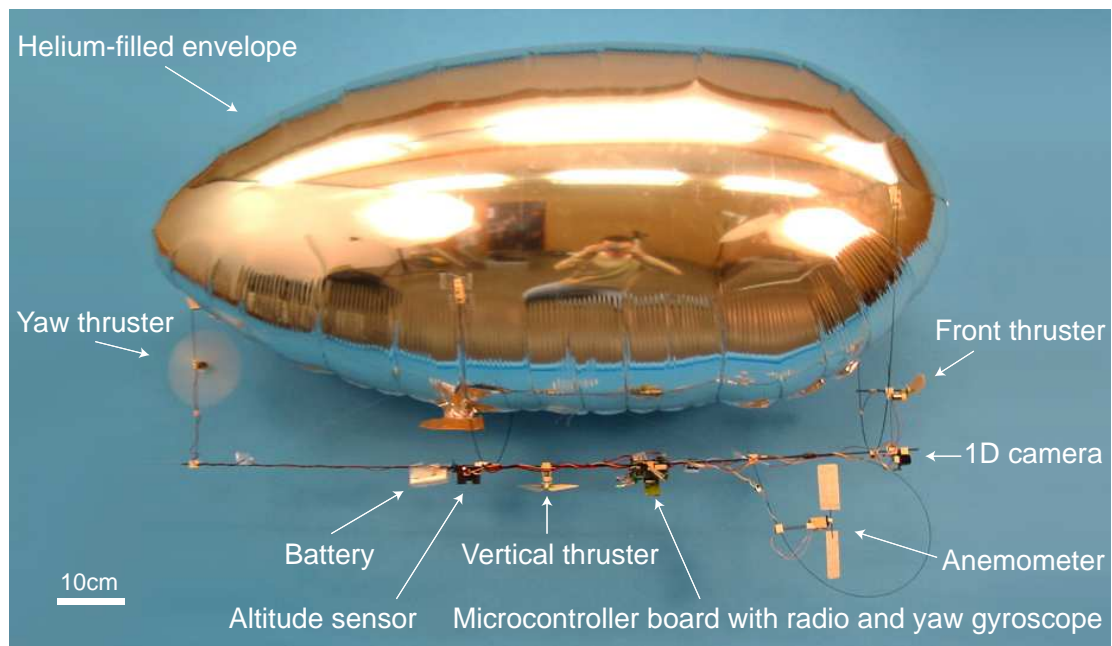


Figure 3.2: Autonomous indoor airship *Blimp2b* with description of all electronic components, sensors and actuators.

Although *Blimp2b* can move in 3D, roll and pitch movements are passively stabilised around the horizontal attitude. Consequently, the *Blimp2b* has virtually only 4 DOF. Furthermore, we implemented an automatic (non vision-based) altitude control using the vertical distance sensor in order to reduce the manoeuvring space to 2D and the control requirements to 3 DOF. Even with this simplification, this airship displays much more complex dynamics than the *Khepera* and no trivial relation exists between the voltages applied to the motors and the resulting trajectory. This is due to inertia (not only of the blimp itself but also of the displaced air in the surroundings of the hull) and to aerodynamic forces (see appendix A). Therefore, thinking in terms of control mechanisms according to section 2.4, in addition to obstacle avoidance (OA), the *Blimp2b* would require course stabilisation (CS) in order to be able to gain forward speed without rotating randomly around its yaw axis. Instead, vision-based altitude control (ALC) is not needed in the case it is ensured by the vertical distance sensor. Because of the natural passive stabilisation, active attitude control (ATC) is not required either.



### 3.1.3 Indoor Airplane

In 2001, together with the EPFL spin-off Didel<sup>TM</sup>, we started the process of developing ultra-light flying airplanes for indoor robotic research (Nicoud and Zufferey, 2002). Rotorcrafts and flapping-wing systems were discarded mainly because of their mechanical complexity, their intrinsic instability and the lack of literature about unsteady-state aerodynamics especially for small size and low speed (i.e., low Reynolds number). Our effort was aimed at a simple platform that should be able to fly in office-like environments, which demand high manoeuvrability and low speed in addition to the need for a small size.

#### Requirements for Flying Indoor

In order to better appreciate the challenges involved, let us review some basic concepts of steady-state aerodynamics. First of all, the lift  $F_L$  and drag  $F_D$  forces acting on a wing of surface  $S$  going through the air at velocity  $v$  are given by:

$$F_{L,D} = \frac{1}{2} \rho v^2 S C_{L,D}, \quad (3.1)$$

where  $\rho$  is the air density and  $C_L$  and  $C_D$  the lift and drag coefficients. Those coefficients depends on the airfoil geometry, its angle of attack and the airflow characteristics around it. The airflow (or any fluid) dynamic characteristics are represented by the dimensionless Reynolds number  $Re$ , which is defined as:

$$Re = \frac{\rho v L}{\mu} = \frac{\rho v^2}{\frac{\mu v}{L}} = \frac{\text{inertial forces}}{\text{viscous forces}}, \quad (3.2)$$

where  $\mu$  is the air dynamic viscosity and  $L$  a characteristic length of the airfoil (generally the average wing chord, i.e., the distance from leading edge to trailing edge).  $Re$  provides a criterion for dynamic similarity of airflows. In other words, two similar objects of possibly different sizes in different fluids with different flowrates will have similar fluid flow around them if  $Re$  is the same in both situations. If the fluid density and viscosity are constant, the Reynolds number is mainly a function of airspeed  $v$  and wing size  $L$ . The Reynolds number also indicates the relative significance of the viscous effect compared to the inertia effect. Obviously,  $Re$  is small for slow-flying, small aerial devices (typically  $0.3\text{-}5 \cdot 10^3$  in flying insects,  $1\text{-}3 \cdot 10^4$  in indoor slow-flyers), whereas it is large for standard airplanes flying faster ( $10^7$  for a Cessna,

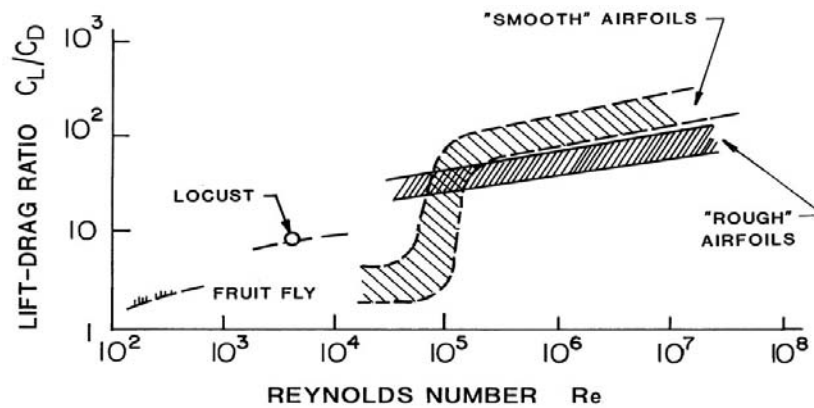


Figure 3.3: Maximum lift-to-drag ratio. Airfoil performance deteriorates rapidly as the Reynolds number decreases below  $10^5$ . Adapted from McMasters and Henderson (1980).

up to  $10^8$  for a Boeing 747). Therefore, very different airflows should be expected between a small and slow flyer and a standard aircraft. In particular, viscous effects are predominant at small size.

The aerodynamic efficiency of an airfoil is defined in terms of the maximum lift-to-drag ratio (Mueller and DeLaurier, 2001). Unfortunately, this ratio has a general tendency to fall off as the Reynolds number decreases (figure 3.3). In addition to flying at a regime of bad aerodynamic efficiency (i.e., low  $C_L$  and high  $C_D$ ), indoor flying robots are required to fly at very low speed (typically 1-2m/s), thus further reducing the available lift force  $F_L$  produced by the wing (equation 3.1). For a given payload, the only way of meeting such constraints is to reach a very low wing-loading ratio (weight to wing surface ratio) by widening the wing surface. Figure 3.4 shows the place of exception occupied by indoor flying robots among other aircraft. It also highlights the fundamental difference between indoor airplanes and micro air vehicles (MAVs) (Mueller, 2001). Although their overall weight is similar, their respective speed ranges are located on both sides of the tendency line. As opposed to indoor flying robots, MAVs tend to have small wings (around 15cm) for ease of packaging and pre-launch handling, and high speed (about 15m/s).

Because of the lack of methods for developing efficient airframe geometries at Reynolds numbers below  $2 \cdot 10^5$  (Mueller and DeLaurier, 2001), we proceeded by trials and errors. Note that even if methods for analytical optimisation of airfoil shape were available, it would be very difficult, if not impossible, to guarantee this shape because of

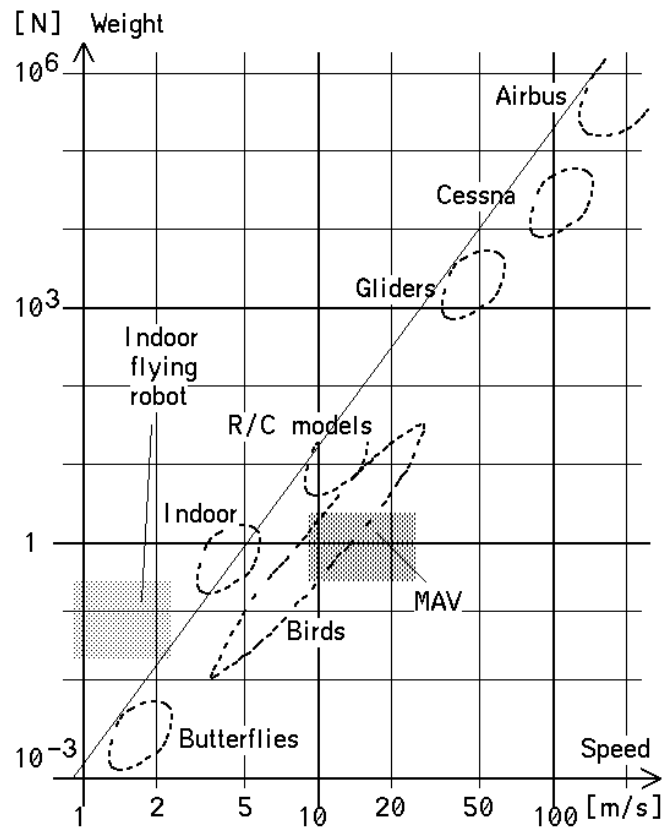
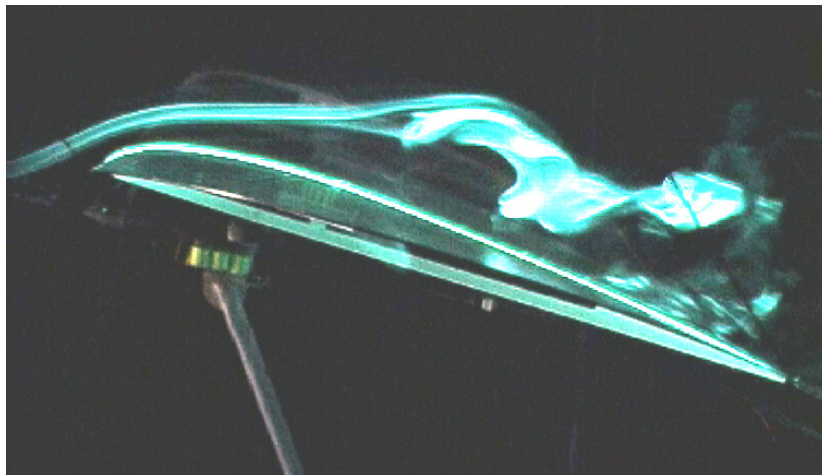


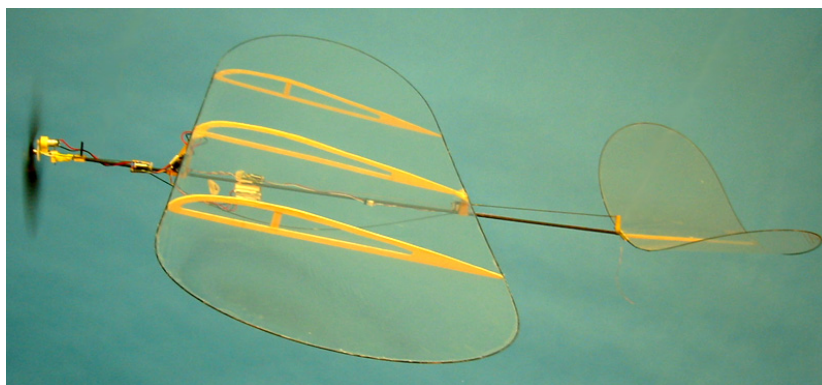
Figure 3.4: Aircraft weight versus speed. “R/C models” denotes typical outdoor radio-controlled airplanes. “Indoor” represents the models used by hobbyists for flying in gymnastic halls. They have less efficiency constraints than “Indoor flying robots” because they can fly faster. “MAV” stands for micro air vehicles (as defined by DARPA). Adapted from Nicoud and Zufferey (2002).

the need for using so lightweight materials. Moreover, in such lightweight systems, the structural parts are so thin that there is always a lot of distortion, especially when flying. This tends to further widen the gap between theoretical and actual airframe geometries. Our approach therefore was rather to concentrate first on feasible airframes that meet the weight constraints and then improve them on the basis of flight tests and wind tunnel experiments.

Our indoor airplanes are made of carbon-fiber rods and balsa wood for the structural part, and of thin plastic film ( $2.2\text{g/m}^2$ ) for the lifting surfaces. Details on wing and propeller design as well as motor selection are given in Nicoud and Zufferey (2002). Tests in wind tunnel allowed us to optimize the wing structure and airfoil by measuring lift and drag for different wing geometries under the same airflow conditions (Zufferey et al., 2001). The measures were obtained with a



(a) Airflow visualisation



(b) The C4 prototype

Figure 3.5: (a) Airflow visualisation over the airfoil of the C4 (shown below) using a smoke-laser technique within a special wind tunnel at low air speed. The prototype is attached to the top of a custom-developed device for measuring very small lift and drag forces. Adapted from Floreano et al. (2005). (b) Preliminary prototype C4 of the indoor airplane.

custom-developed aerodynamic scale capable of detecting very weak forces and torques. Furthermore, by employing visualisation techniques (figure 3.5a), we were able to analyze suboptimal airflow conditions and modify the airframe accordingly.

Since 2001, various prototypes have been developed and tested. The first operational one was the C4 (figure 3.5b). Weighing 47g without any sensors (see Zufferey et al., 2001, for the weight distribution), this 80cm-wingspan airplane was able to fly between 1.4 and 3m/s with a turning radius of approximately 2m. The batteries used at that time

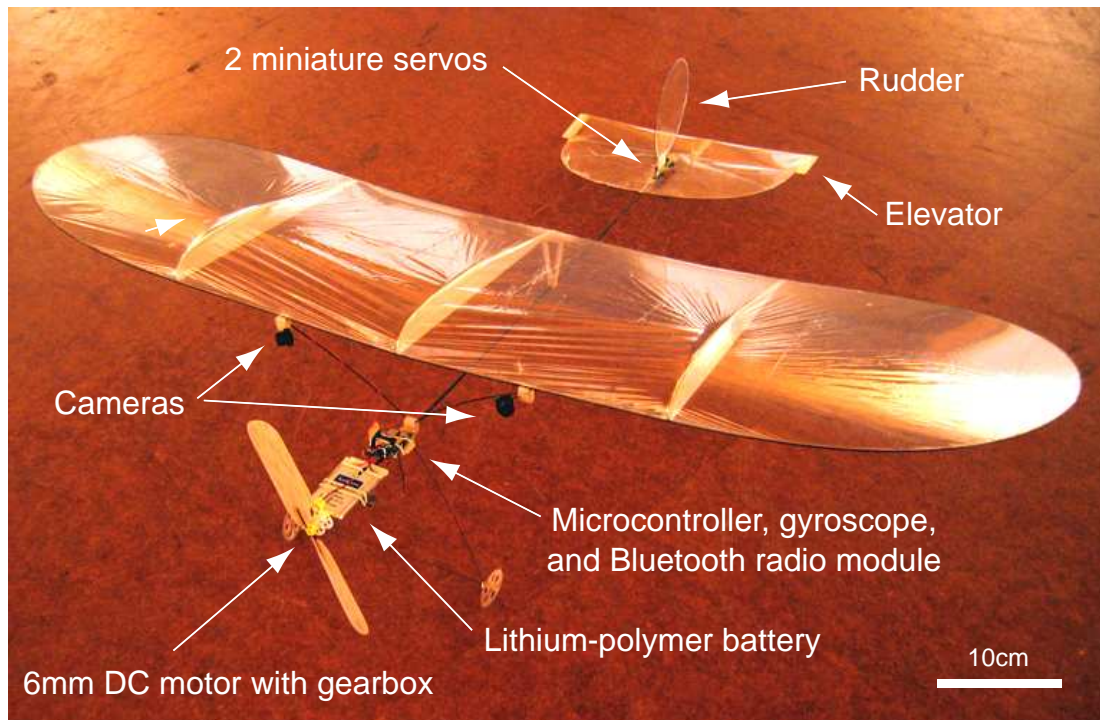


Figure 3.6: Autonomous indoor airplane *F2* with description of all electronic components, sensors and actuators.

provided an autonomy of a mere 5 minutes.<sup>1</sup>

### The F2

The latest version of our robotic indoor flyers, the model *F2* (figure 3.6), was specifically developed for the experiment of autonomous steering reported in chapter 5. The *F2* has a wingspan of 86cm and an overall weight of only 30g including two vision sensors and a gyroscope (table 3.1). Thanks to its very low inertia, the *F2* is hardly damaged when crashing into obstacles. This characteristic is mostly appreciated during early phases of controller development. In order to further limit the risk of damaging the aircraft, the walls of the experimental arena used for this robot are made of fabric (section 3.4).

The *F2* flight speed lies between 1.2 and 2.5m/s and its yaw angular rate is in the  $\pm 100^\circ/\text{s}$  range. At 2m/s, the minimum turning radius is less than 1.3m. The *F2* is propelled by a 6mm DC motor with a gearbox driving a balsa-wood propeller. Two miniature servos

<sup>1</sup>The realisation of the first prototypes (among which the *C4*) as well as the wind-tunnel experiments were carried out together with Cyril Halter and with the help of Jean-Daniel Nicoud for the realisation of the aerodynamic scale.

Fuselage and tail	: 4.7g
Wing	: 4.5g
Landing gear	: 1.2g
Motor, gearbox, propeller	: 2.7g
Two servos	: 2.7g
Battery	: 6.9g
Microcontroller board with gyroscope	: 3.0g
Two 1D cameras with optics	: 2.0g
Bluetooth radio module	: 1.0g
Miscellaneous (cables, glue)	: 1.3g
Total	: 30g

Table 3.1: Weight distribution of the F2.

(GD-servo from Didel™) are placed at the back end of the fuselage to control the rudder and elevator. On-board energy is provided by a 310mAh Lithium-polymer battery. The power consumption of the electronics (including wireless communication, see subsection 3.2.3) is about 300mW, whereas overall peak consumption reaches 2W. The energetic autonomy in flight is around 30 minutes.

In order to provide this airplane with good passive stability around roll and pitch angles, the wing is positioned rather high with respect to the center of gravity and the tail is located relatively far behind the wing. In addition, some dihedral<sup>2</sup> naturally appears in flight because of the distortion of the longitudinal carbon rods holding the wings. This effect also contributes to passive roll stability. As a result, no active attitude control (ATC) is actually needed in order for the F2 to stay upright in flight. Course stabilisation (CS) can be useful to counteract air turbulences and the effects of airframe asymmetries. Obstacle avoidance (OA) is the central issue of chapter 5, whereas altitude control (ALC) has not yet been tested on this airplane.

### 3.1.4 Comparative Summary of Robotic Platforms

Table 3.2 provides an overview of the three robotic platforms described above. The first part of the table summarises their main characteristics. The second part is about electronics and sensors that will be

<sup>2</sup>Dihedral is the upward angle of an aircraft's wings from root to tip, as viewed from directly in front of or behind the aircraft. The purpose of dihedral is to confer stability in the roll axis. When an aircraft with dihedral is yawing to the left, the dihedral causes the left wing to experience a greater angle of attack, which increases lift. This increased lift tends to cause the aircraft to then return to level flight.

	Khepera with kevopic	Indoor airship (Blimp2b)	Indoor airplane (F2)
Type	Terrestrial, wheeled	Aerial, buoyant	Aerial, fixed-wings
Degrees of freedom (DOF)	3	4	6
Actuators	2 wheels	3 propellers	1 propeller + 2 servos
Weight [g]	120	180	30
Speed range [m/s]	0 to 0.2	0 to 1	1.2 to 2.5
Typical arena size [m]	0.6 x 0.6	5 x 5	16 x 16
Typical power consumption [W]	4	1	1.5
Power supply	cable	battery	battery
Energetic autonomy	-	2-3 hours	15-30 minutes
Microcontroller board	kevopic	bevopic	pevopic
Vision sensor	1x	1x	2x
Yaw gyroscope	1x	1x	1x
Additional sensors (not used by the navigation controller)	wheel encoders, 8 proximity sensors	anemometer, vertical range finder	-
Control mechanisms theoretically required for autonomous operation	OA	CS, OA, (ALC)	CS, OA, ALC
Hand-crafted control mechanisms (chapter 5)	OA, ALC (as wall following)	-	CS, OA
Support evolutionary experiments (chapter 6)	yes	yes	no

Table 3.2: Characteristics of the three robotic platforms

described in the next section. The last lines anticipate the two main chapters of this dissertation by summarising which control mechanisms (section 2.4) will be tested on which robot in chapter 5 and which robots can be and will be engaged in evolutionary experiments in chapter 6.

Note that this set of three platforms feature increasing dynamic complexity, speed range, and degrees of freedom. This allows to assess control strategies and methodologies with an incremental degree of complexity (Zufferey et al., 2003).

## 3.2 Unified Embedded Electronics

The electronic equipment of the different robots was conceived to facilitate the transfer of technology and software from one platform to another. In this section, we present the microcontroller boards, the sensors, and the communication systems equipping the three robotic platforms.

### 3.2.1 Microcontroller Boards

We developed three similar microcontroller boards (figure 3.7) for each of the three platforms. They can be programmed using the same tools and software modules can easily be exchanged. A common aspect of these boards is that they are all based on a Microchip™ 8-bit microcontroller. The PIC18F6720 microcontroller has been selected as the core processor for different reasons. First it consumes only 30-40mW when running at 20MHz. Then, it also supports low voltage (3V) power supply, which is compatible with one Lithium-polymer cell (3.7V nominal). It is available in very small packaging (12x12x1mm, Thin Quad Flat Package) and has therefore a minimal weight (less than 0.3g). Furthermore, it features a number of integrated hardware peripherals like, e.g., USART (Universal Synchronous Asynchronous Receiver Transmitter), MSSP (Master Synchronous Serial Port, in particular I2C), and ADCs (Analog to Digital Converters) allowing different types of interfaces with the robot sensors and actuators. The microcontroller can be programmed in assembler as well as in C-language (using, e.g., the CCS™ PICC compiler), which enhances the code readability, portability, and modularity.

Obviously, the advantages such as low power consumption and small size come at the expense of other features. The PIC18F6720 has a reduced instruction set (e.g., 8-bit addition, multiplication, but no division), does not support floating point arithmetic, and features limited memory (3840 bytes of RAM, 64k words of program memory). However, in our approach to controlling indoor flying robots, the limited available processing power is taken as a typical constraint of such platforms. Therefore, the majority of navigation experiments, at least in their final stage, are performed with embedded software in order to demonstrate the adequacy of the proposed control strategies with autonomous indoor flying robots.

The microcontroller board for the Khepera, the so-called *kevopic*,



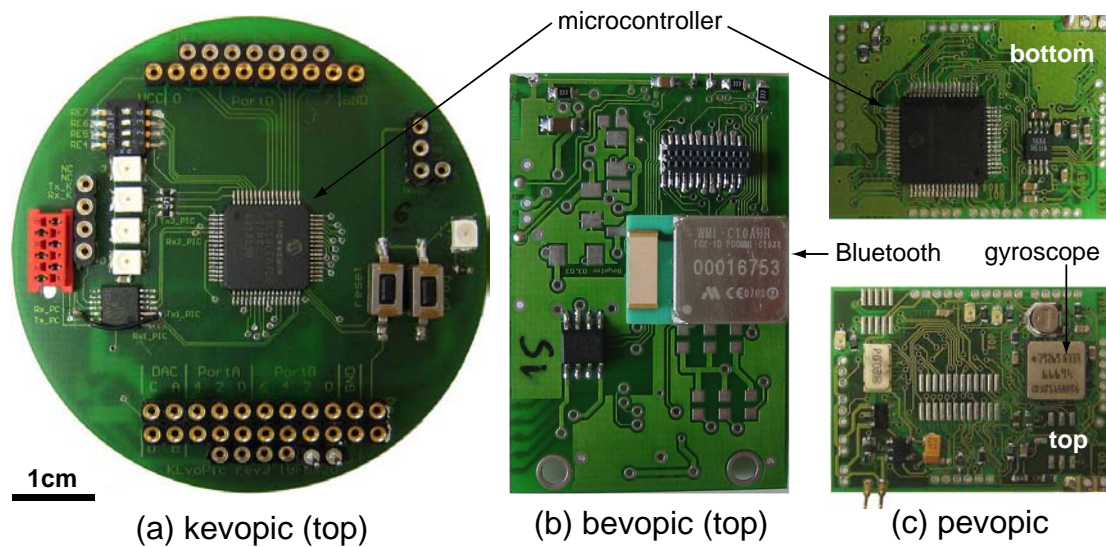


Figure 3.7: Microcontroller boards (a) *kevopic* (for the Khepera), (b) *bevopic* (for the blimp) and (c) *pevopic* (for the plane). The microcontroller of the *bevopic* is on the bottom side of the board (not visible in the picture). The Bluetooth™ module with its ceramic antenna (beige rectangle) is shown only on *bevopic*, but is also used with *pevopic*. Also visible on the *pevopic* is the gyroscope (subsection 3.2.2).

is not directly connected to some of the robots peripherals (motors, wheel encoders, and proximity sensors), but uses the underlying Khepera module as a slave. *Kevopic* has only a serial communication link with the underlying Khepera, which is only employed for sending motor commands, reading wheel speeds and proximity sensors. The visual and gyroscopic sensors, instead, are directly connected to *kevopic*, avoiding the transfer of vision stream via the Khepera main processor.

The architecture is slightly different for the two next boards for the flying robots because they are directly interfaced with both the sensors and actuators. In addition to the PIC microcontroller, *bevopic* (blimp, evolution, PIC) features three motor drivers and plenty of extension connectors, including one for the vision sensor, one for the gyroscope, and one for the other sensors and actuators. It is slightly smaller and far lighter than *kevopic* (4.4g instead of 14g). It also features a connector for a Bluetooth™ radio module (subsection 3.2.3).

The latest rendition of the microcontroller board for the F2 airplane, *pevopic*, is largely similar to *bevopic*, although it features lower weight and smaller size. *Pevopic* weighs only 4g, including the wireless module, and is twice as small as *bevopic* (figure 3.7). This was possible because the plane does not require bidirectional motor drivers. A sim-

ple transistor is sufficient for the main motor (servos for rudder and elevator have their own motor drivers). Unlike bevopic, pevopic has the gyroscope directly onboard in order to spare the weight of the connection wires and additional electronic board.

### 3.2.2 Sensors

As in flies, vision and gyroscope are the core sensory modalities of our robots. Fortunately, it was possible to use the same vision chip and the same gyroscope on all three robotic platforms. Modifications were only required on optics and packaging in order to meet the different constraints of the robotic platforms.

#### Camera and Optics

The selection of a suitable vision system that should be able to provide enough information about the surrounding world for autonomous navigation while fitting the drastic weight constraints of a 30-gram flying robot (on which only 2-3 grams are left for visual sensors) is not a trivial task. On the one hand, it is well known that global motion fields spanning wide field of view (FOV) are easier to interpret (Nelson and Aloimonos, 1988) and indeed most flying insects have almost omnidirectional vision (subsection 2.3.3). On the other hand, artificial vision systems with wide FOV are likely to be too heavy because they need either a special mirror or fish-eye optics with multiple glass-lenses. Such subsystems are also likely to require much, if not too much, processing power from the onboard microcontroller.

Therefore, we decided to use simple, low-resolution, and lightweight 1D cameras (also called *linear* cameras) with lightweight plastic lenses pointing at relevant directions according to the behaviour (see chapter 5). 1D cameras also present the advantage of having few pixels, hence keeping the computational and memory requirements within the limits of a simple microcontroller. The 1D camera we selected for the experiments reported in this dissertation is the Taos Inc. TSL3301 (figure 3.8). It features a linear array of 102 grey-level pixels, out of which only 50 pixels in the middle are exposed to light by the lens.

Before deciding in favour of this 1D camera, we tested other small and lightweight CMOS<sup>3</sup> sensors (Zufferey et al., 2003). Among

---

<sup>3</sup>A major advantage of CMOS (Complementary Metal Oxide Semiconductor) over CCD (Charge Coupled Device) technology is the ability to integrate additional cir-



Figure 3.8: Vision chip (bottom-left), optics (top) and camera packaging (bottom center and right). Marshall and EL-20 optics are interchangeable in the camera for kevopic. In an effort of miniaturisation, the TSL3301 has been machined such to fit the small custom-developed package labelled “Camera for the F2”, whose overall size is only 10x10x8mm. The 8 pins of the TSL3301 have been removed and the chip has been directly soldered on the underlying printed circuit board. The EL-20 core plastic lens has been extracted from its original packaging and placed into a smaller one (top-right). The weight gain is fivefold (a camera for kevopic with an EL-20 weighs 4g).

them was an interesting 2D camera module, named OV7645FB, from Omnivision<sup>TM</sup>. This module would have been sufficiently small (10x9x7mm) and lightweight (0.7g) to be mounted on the F2. The core chip is a VGA color camera, in which acquisition can be restricted to a sub-region of the whole image (windowing). An 8-bit parallel bus is used to transfer pixel values while a serial port allows for adjustment of camera parameters such as brightness, contrast, gain, windowing, etc. Despite the remarkable capabilities of this camera module, the OV7645FB tends to output too many data (more than 300'000 pixels) for the microcontroller (whose data memory has only 3840 bytes).

cuitry on the same die as the sensor itself. This makes it possible to integrate analog to digital converters or other functionalities like windowing. Furthermore, CMOS imagers offer lower power dissipation, and smaller system size than CCD sensors. These advantages are at the expense of image quality but this criterion is of minor importance in our case, especially in comparison with the coarse resolution of insects eyes.

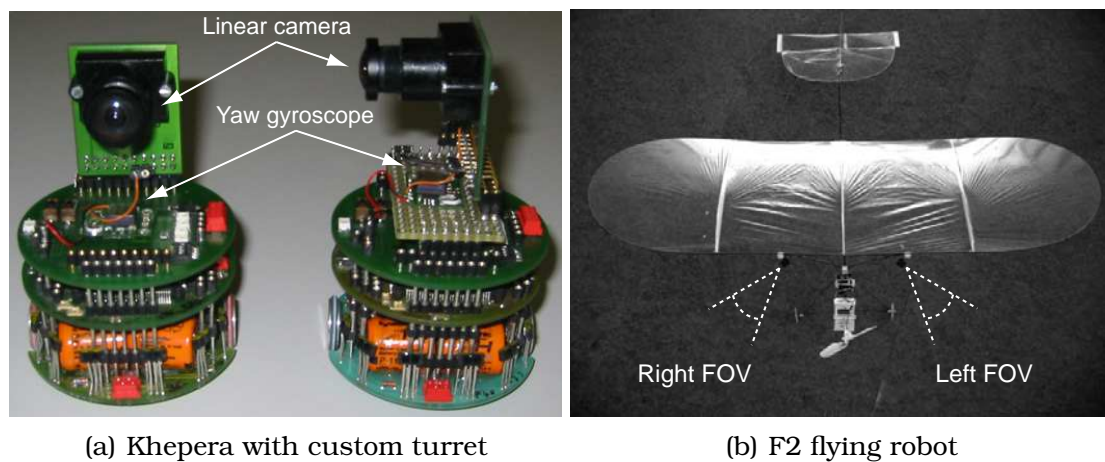


Figure 3.9: Camera position and orientation on the robots (the blimp case is not shown). (a) On the Khepera, the camera can be oriented either forward or laterally with  $70^\circ$  or  $130^\circ$  FOV depending on the optics (on this picture, the Marshall  $120^\circ$  lens is mounted). (b) Top view of the F2; camera orientations are indicated in white dashed lines.

Although a large part of those pixels could be ignored if only low resolution images are acquired, the camera still needs to scan every pixel internally. Therefore, it is very difficult to obtain high frame rates while maintaining a sufficiently slow pixel clock for the microcontroller to be able to read and store the pixels arriving on its parallel port. The maximum possible frame rate compatible with the PIC was 12.5Hz, which is quite slow when compared to the 300Hz of the fly's eye (see subsection 2.2.1). This slow frame rate would precludes optic-flow algorithms to provide precise information. For sake of comparison, the maximal frame rate of the TSL3301 is around 1kHz.

### Optics and Camera Orientations

In order to focus the light onto the TSL3301 pixel array, we use two different optics (figure 3.8). The first one, a Marshall-Electronics™ V-4301.9-2.0FT, has a very short focal length of 1.9mm providing an ultra-large FOV of about  $120^\circ$  at the expense of a relatively large weight of 5g. The second one, an Applied-Image-group™ EL-20, has a focal length of 3.4mm and a FOV of approximately  $70^\circ$ . The advantages of the EL-20 are its relatively low weight (1g) due to its single plastic lens system and the fact that it could be machined in order to extract the core lens and remount it in a miniaturised lens-holder weighing only 0.2g (figure 3.8, top-right). Both optics provide an interpixel angle (1.4-

2.6°) comparable to the interommatidial angle in flying insects (1-5°, see subsection 2.2.1).

In all experiments reported in this thesis, the TSL3301 line of pixels is oriented horizontally. On the Khepera, the camera can be oriented either forward or laterally by adding a small adapter (figure 3.9). On the Blimp2b, the camera is mounted at the front end of the gondola and oriented forward (figure 3.2). For the experiment of chapter 5, the F2 airplane needs a large FOV, but the weight of a Marshall lens is prohibitive. In fact, the Khepera and the Blimp2b support both types of lenses, whereas the F2 is equipped with two miniaturised camera modules each oriented at 45° off the longitudinal axis of the plane (figure 3.9), and featuring the EL-20 core lens. Two miniature cameras with custom packaging are indeed 10 times lighter than a single one with a Marshall lens (figure 3.8).

### **Gyroscope**

Looking for a very small and lightweight gyroscope with minimum external components, we found the Analog-Devices™ ADXRS (figure 3.10), which is an affordable, self-contained MEMS (Micro-Electro-Mechanical Systems) gyroscope. It is mounted on a ball-grid array package, 7mm square and 3mm thick. It consumes only 25mW but requires a small step-up converter to supply it with 5V (instead of 3.3V for the rest of the on-board electronics).

Very much like the halteres of the flies (subsection 2.2.2), such piezoelectric gyroscopes use the Coriolis effect on vibrating elements to sense the speed of rotation. The ADXRS150 can sense angular velocity up to 150°/s. Taking into account the analog to digital conversion done in the microcontroller, the resolution of the system is slightly better than 1°/s over the whole range. Each of our three robots are equipped with one gyroscope to measure yaw rotation. The one on the Khepera is visible in figure 3.9. The one on the Blimp2b is shown in figure 3.2. The one for the F2 is directly mounted on the pevopic board and shown in figure 3.7.

### **3.2.3 Communication**

It is crucial to have a communication link supporting bidirectional data transfer in real-time, in order to monitor the robot's internal state during experiments. In this respect, the Khepera is very practical be-

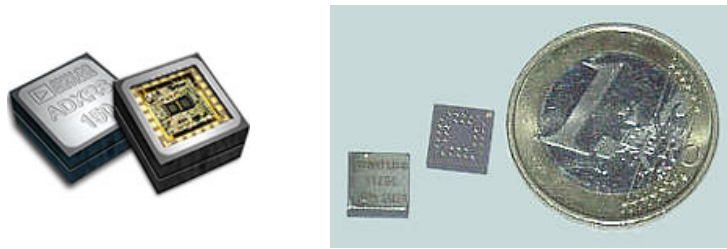


Figure 3.10: The ADXRS piezoelectric gyroscope. The package is 7mm square and 3mm thick and weighs 0.4g.

cause it can easily be connected to the serial port of a workstation with wires through a rotating contact module (as shown in figure 3.12a). Of course, this is not possible with the aerial versions of our robots. To meet the communication requirements, we opted for a Bluetooth<sup>4</sup> wireless solution. Commercially available Bluetooth radio modules can be connected directly on an RS232 serial port. We developed basic drivers for the PIC and the PC allowing to communicate with such Bluetooth modules at the HCI level (Host Controller Interface). This level does not provides full compatibility with commercially available Bluetooth USB devices, but bypasses higher level protocol stacks so that overheads and delays are kept to a minimum.

The selected Bluetooth module, a Mitsumi<sup>TM</sup> WML-C10-AHR (shown on the bevopic board in figure 3.7), has a built-in ceramic antenna for an overall weight of only 1g. It is a class 2 module, meaning that the communication range is guaranteed up to 10m, but in practice distances up to 25m in indoor environments are not a problem with that particular module. The power consumption is 100 to 150mW during transmission.

The advantages of using Bluetooth technology are twofold. First, in the long run we benefit from the continuous efforts toward low power and small modules. Second, Bluetooth modules have several built-in mechanisms against electromagnetic noise, such as frequency hopping and automatic packet retransmission on errors. Therefore, the host microcontroller does not need to worry about encoding or error detection and recovery.

To communicate with the robots, we use a simple packet-based communication protocol. The packets can be encapsulated into HCI<sup>5</sup> packet when using Bluetooth or simply sent over an RS232 channel

<sup>4</sup><http://www.bluetooth.org>

<sup>5</sup>Host-Controller Interface



when working with a cable. Bevopic and pevopic have indeed a connector supporting either a RS232 cable or a Bluetooth module. When Bluetooth is used, the PIC controls the module via the same serial port. Note that a packet-based protocol is also very convenient for TCP/IP communication, which we employ when working with simulated robots (subsection 3.3.2).

## 3.3 Software Tools

This section briefly discusses the two main software tools that are used for the experiments of chapters 5 and 6. The first one is a robot interface and artificial evolution manager used for fast prototyping of control strategies and for evolutionary experiments. The second software is a robot simulator that is mainly used for the Blimp2b.

### 3.3.1 Robot Interface

The software *goevo*<sup>6</sup> is a robot interface written in C++ with the wxWidgets<sup>7</sup> framework, to ensure multi operating systems compatibility. Goevo implements the simple packet-based protocol (subsection 3.2.3) over different kinds of communication channels (RS232, Bluetooth, TCP/IP) in order to receive and send data from/to the robots. It can display sensor data in real-time and log them into text files that can be further analysed with a mathematical software. It is also very convenient for early stage assessment of sensory-motor loops since control schemes can be easily implemented and assessed on a workstation (which communicates with the real robot at every sensory-motor cycle) before being compiled into the microcontroller firmware for autonomous operation.

Goevo can also be used to evolve neural circuits for controlling real or simulated robots. It features built-in neural networks and an evolutionary algorithm (chapter 6). We do not describe this software in more details here because it has been collectively developed and is available online in open-source format.

---

<sup>6</sup>goevo website: <http://lis.epfl.ch/resources/evo>

<sup>7</sup>wxWidgets website: <http://wxwidgets.org/>

### 3.3.2 Robot Simulator

A robot simulator is also used to ease the development of control strategies before validating them in reality. This is particularly useful with evolutionary techniques (chapter 6) that are known to be time consuming when performed in reality and potentially destructive for the robots.

As a framework for simulating our robots, we employ Webots<sup>TM</sup> (Michel, 2004), which is a convenient tool for creating and running mobile robot simulations in 3D environments (using OpenGL) with a number of built-in sensors like cameras, gyroscopes, bumpers, range finders, etc. Webots also features rigid-body dynamics (based on ODE<sup>8</sup>). The dynamics engine provides libraries for kinematic transformations, collision handling, friction and bouncing forces, etc. Goevo can communicate with a robot simulated in Webots via a TCP/IP connection, using the same packet-based protocol as employed with the real robots.

The Khepera robot with its wheel encoders and proximity sensors is readily available in the basic version of Webots. For our purpose, it has been augmented with a 1D vision sensor and a gyroscope to emulate the functionality provided by kevopic. The experimental arena is easy to reconstruct using the same textures as employed to print the wallpaper of the real arena.

Webots does not yet support non-rigid-body effects such as aerodynamic or added-mass effects. In order to ensure realistic simulation of the Blimp2b, we added the dynamic model presented in Appendix A as a custom dynamics of the simulated robot, while leaving it to Webots to handle friction with walls and bouncing forces when necessary. The custom dynamics implementation takes current velocities and accelerations as input and provides force vectors that are passed to Webots, which computes the resulting new state after a simulation step. Figure 3.12d illustrates the Blimp2b in its simulated environment in Webots. The simulated Blimp2b features the same set of sensors as its real counterpart (figure 3.11). Those sensors are modeled using data recorded from the physical robot. Noise level and noise envelope were reproduced in the simulated sensors so to match as closely as possible the real data. In addition to the sensors existing on the physical Blimp2b, *virtual sensors*<sup>9</sup> can easily be implemented in simulation.

---

<sup>8</sup>Open Dynamics Engine website: <http://opende.sourceforge.net>

<sup>9</sup>We call “virtual sensors” the ones that are only implemented in simulation, but do not exist on the real blimp.



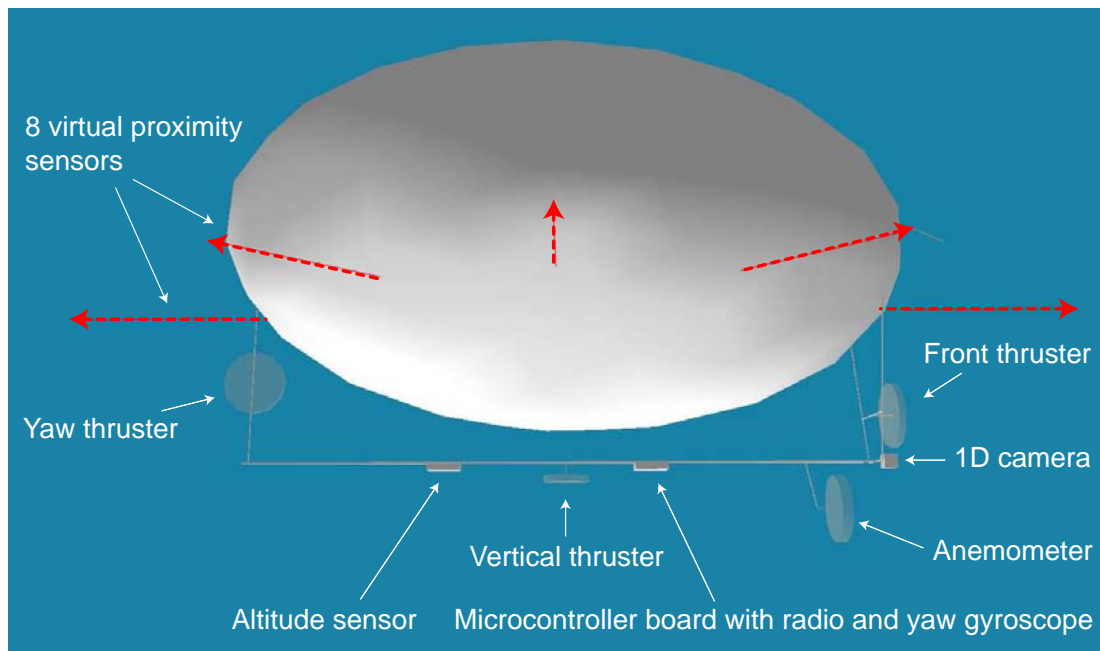


Figure 3.11: Side view of the simulated Blimp2b. The dotted thick arrows indicate the direction and range of the virtual proximity sensors.

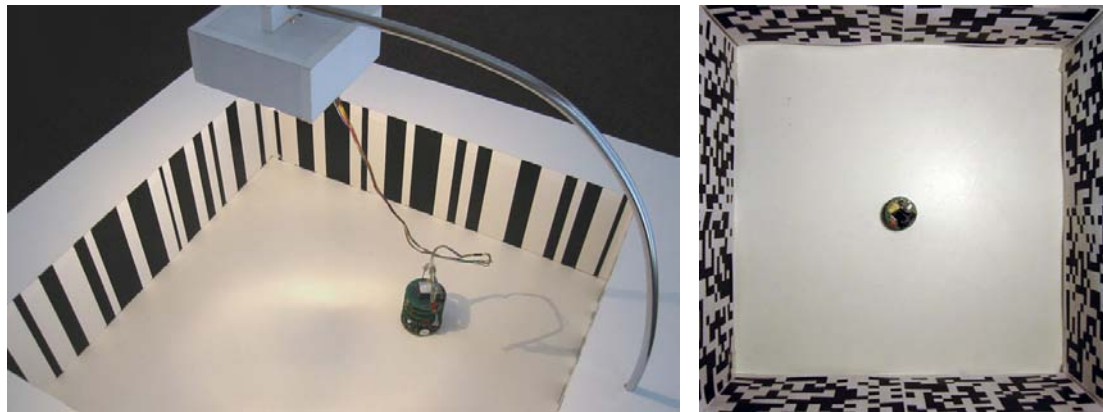
In particular, for the need of experiments performed in chapter 6, the simulated Blimp2b is provided with 8 proximity sensors distributed all around the envelope (figure 3.11).

A preliminary dynamic model of the F2 has been developed (Guanella, 2004) but its accuracy has not yet been thoroughly measured. Therefore, experiments with the simulated airplane are not covered in this dissertation.

The simulation rate obtained with all sensors enabled and full physics (built-in and custom) is 40 to 50 times faster than real-time when running on a current PC (e.g., Intel(R) Pentium IV at 2.5GHz with 512MB RAM and nVidia(R) GeForce4 graphic accelerator). This rate allows to significantly accelerate long-lasting experiments such as evolutionary runs.

### 3.4 Experimental Environments

Since this thesis is focused on simple, vision-based navigation strategies, the geometry of the experimental arenas is deliberately kept as simple as possible (figure 3.12). The square textured arenas are inspired by the environments that are used in biology for studying



(a) Khepera arena

(b) With another pattern



(c) Real blimp arena

(d) Simulated blimp arena



(e) Airplane arena

Figure 3.12: Experimental environments. (a) The Khepera arena is 60x60cm with 30cm high walls featuring randomly arranged black and white patterns. (b) The same square arena for the Khepera with another kind of random pattern on walls. (c) The blimp arena measures 5x5m and has the same kind of random black and white stripes painted on the walls. (d) The same arena but in simulation. The patterns in the simulator are exactly reproduced from the real ones that have been painted on the walls. (e) The larger test arena for the 30-gram indoor airplane. This one is 16x16m large and delimited by soft walls made of fabric. Note that the regularity of the pattern is due to the size of the material from the store, but is not required by the aircraft.

visually-guided behaviours of insects (see, e.g., Egelhaaf and Borst, 1993a; Srinivasan et al., 1996; Tammero and Dickinson, 2001, or figure 2.7). Although in biological experiments arenas are often cylindrical rather than square, the simplicity of shape and texture facilitates the understanding of the principles underlying insects' behaviour and development of the robot control systems.

A constraint specific to our incremental approach using different types of robots is also to have similar environments with different sizes adapted to each platform's velocity and dynamics. Available infrastructure has also to be taken into account in this choice, since it could be difficult to find convenient rooms in the vicinity of the laboratory. Taking care of those constraints, we decided to go for square rooms, whose dimensions are roughly adapted to the size and manoeuvrability of the robots. The Khepera has a small desktop arena of 60x60cm, the blimp manoeuvres in a room measuring 5x5m (3m high), and the indoor airplane flies in a 16x16m arena delimited by fabric walls.

In order to provide robots with visual contrast, the walls of those arenas are equipped with random black and white patterns. The random distribution and size of the stripes is intended to make sure that no trivial geometrical solutions to depth perception could be used by the robots to navigate. The same kind of random distribution was too expensive to obtain in the wide arena for the airplane (figure 3.12e) because fabric size was standard and we could not spend time to cut and reassemble the pieces of fabric. However, even with this almost homogeneously distributed patterns, the robot does not rely on static geometry to navigate in this room (chapter 5).

The kind of walls and the way patterns are applied on them depends on the type of experiments and robot characteristics. The Khepera arena has been made such as to be easily covered with different wallpapers (e.g., figure 3.12b). In this case, the patterns are directly printed on paper sheets and glued on the walls. It was impossible to apply the same technique for the blimp because of the greater dimensions of the room. Therefore, the black stripes were directly painted over white walls. Finally, the airplane arena is made of soft walls not only because they are easily mounted and dismantled, but also because they absorb energy if a crash occurs during development.

### 3.5 Summary and Discussion

In this chapter, we presented the robotic platforms that are used for vision-based navigation experiments together with the tools and experimental arenas. The Khepera with kevopic is the simplest from a dynamic and operational perspective because it moves on a flat surface and in a limited space, and can be wired to a computer without affecting its dynamics. Therefore, it is used for preliminary assessment of control strategies. However, this wheeled platform does not capture the complex dynamics of flying robots. In this thesis, the targeted platform is the F2 airplane because it features the lightest weight and has a dynamics, which is the closest to a flying insect.<sup>10</sup> This robot is thus used in chapter 5 to demonstrate autonomous, optic-flow-based aerial steering. Looking for alternative, vision-based navigation strategies, chapter 6 relies on an evolutionary techniques, which presents several difficulties for a robot as the F2. Therefore, we took the strategic decision to tackle this experimental approach with a more convenient testbed before extending it to winged platforms (not yet achieved). To that end, we developed the Blimp2b, which can fly more easily than the F2 and is able to withstand chocks without damage. It also has a simpler dynamic model than a plane because critical situations such as stall or aerobatic manoeuvres do not occur with airships. Therefore an accurate simulation of a blimp is simpler to obtain and has been successfully developed and integrated in our robotic simulator (Webots<sup>TM</sup>). As a side effect, the hovering and backward flying capabilities of the blimp allows to investigate behaviours that could be reused on other hovering platforms such as helicopters.

Obviously, the two aerial platforms do not attempt to reproduce the bio-mechanical principles of insect flight. The choice of using non-biomorphic platforms is also strategic. Although flapping-wings (with even smaller size, see subsection 1.2.1) are likely to provide a good solution for flying within cluttered environments, their aerodynamics of flight is so complex (and not yet fully understood) so that the project would have focused primarily on mechanical and aerodynamics aspects, at the expense of the control strategies. In the near future, there is still lots of work to do on the control side of indoor flying robots, using more classical airplane configurations. As an example, the latest remote-controlled model from Didel<sup>TM</sup>, the so-called *miniCeline*,

---

<sup>10</sup>For instance, its velocity is in the same range as most Diptera, i.e., around 1-3m/s (Dudley, 2000, p.76).

reaches an autonomy of 15 minutes with a 60mAh Lithium-polymer battery. It weighs only 7g and features a minimum airspeed of less than 1m/s for a wingspan of 42cm. Such a low flight speed allows for manoeuvring in cluttered office-like environments, for flying along corridors and even through doors.



# Chapter 4

## Optic Flow

The real voyage of discovery lies not in seeking new landscapes, but in having new eyes.

---

M. Proust (1871-1922)

### Abstract

The main sensory cue in flight control of insects seems to be visual motion (chapter 2), also called *optic flow* (OF). In this chapter we define it more precisely. The formal description of OF will enable to plot the global motion fields generated by particular movements and surrounding structures in order to analyse them and develop the control strategies presented in the next chapter. In practice, lightweight robots cannot afford for high-resolution, omnidirectional cameras and computationally intensive algorithms. OF has to be estimated with limited resources in terms of processing power and vision sensors (see section 3.2). In the second part of this chapter, we thus describe an algorithm for OF detection that meets the constraints imposed by the embedded 8-bit microcontroller. Combined with the 1D camera, this algorithm results in what we call an *optic flow detector* (OFD).<sup>1</sup> This OFD is capable of measuring in real-time image motion along one direction in a selectable part of field of view. Several of those OFDs spanning one or more cameras will be implemented on the robots to serve as image preprocessing for navigation control.

---

<sup>1</sup>Here we use OFD instead of EMD in order to highlight the significant differences that will be further described in subsection 4.2.1.

## 4.1 What is Optic Flow?

Optic flow is the perceived visual motion of objects as the observer moves relative to them. It is generally very useful for navigation because it contains information about self-motion and 3D structure of the environment. The fact that visual perception of changes represents a rich source of information about the world has been widely spread by Gibson (1950). In this thesis, we assume a stationary environment, so that the optic flow is always solely generated by self-motion of the observer.

### 4.1.1 Motion Field and Optic Flow

In general, a difference has to be made between *motion field* (sometimes also called *velocity field*) and *optic flow* (or *optical flow*). The motion field is the 2D projection onto a retina of the relative 3D motion of scene points. It is thus a purely geometrical concept, which has nothing to do with image intensities. On the other hand, the optic flow is defined as the apparent motion of the image intensities (or brightness patterns). Ideally, the optic flow will correspond to the motion field, but this needs not always be the case (Horn, 1986). The principal reasons for discrepancies between optic flow and motion field are the possible absence of brightness gradients or the aperture problem<sup>2</sup>.

In this thesis, however, we deliberately confound this two notions. In fact, there is no need, from a behavioural perspective, to rely on the ideal motion field. It is sufficient to know that the perceived optic flow tends to follow the motion field's main characteristics (such as increase when closing in on objects). This is very likely to be the case in our experimental environments where a lot of visual contrast is available (section 3.4). Moreover, spatial and temporal averaging is used (as in biological systems) to smooth out perturbations arising in small parts of the visual field where no image patterns would be present for a short period of time.

In addition, there is always a difference between the actual optic flow arising on the retina and the one a specific algorithm will measure. However our simple robots are not intended to retrieve metric information about the surrounding world, but rather use qualitative

---

<sup>2</sup>If the motion of an oriented element is detected by a unit that has a small FOV compared to the size of the moving element, the only information that can be extracted is the component of the motion perpendicular to the local orientation of the element (Marr, 1982, p.165, Mallot, 2000, p.182).



properties of optic flow to navigate. Relying on rough optic flow values for achieving efficient behaviours rather than trying to estimate accurate distances is indeed what flying insects are believed to do (Srinivasan et al., 2000). There is also good evidence that flies do not solve the aperture problem, at least at the level of the tangential cells (Borst et al., 1993).

In section 5.1, we shall use the formal description of the motion field in order to build ideal optic flow fields arising in particular flight situations and draw conclusions about the typical flow patterns that can be used for implementing basic control strategies like obstacle avoidance (OA) and altitude control (ALC). Since unlike the eyes of flying insects, the cameras of our robots have limited FOV (subsection 3.2.2), this qualitative study also provides a basis for deciding in which directions the cameras (and thus the OFDs) should be oriented.

### 4.1.2 Formal Description and Properties

Here we discuss the formal definition of optic flow (as if it were identical to the motion field) and highlight interesting properties.

A vision sensor moving within a 3D environment ideally produces a time-varying image which can be characterised by a 2D vector field of local velocities. This motion field describes the 2D projection of the 3D motion of scene points relative to the vision sensor. In general, the motion field depends on the motion of the vision sensor, the structure of the environment (distances from objects), and the motion of objects in the environment, which we assume to be null in our case (stationary environment).

For the sake of simplicity, we consider a spherical visual sensor of unit radius<sup>3</sup> (figure 4.1). The image is formed by spherical projection of the environment onto this sphere. Apart from resembling the case of a fly's eye, the use of a spherical projection makes all points in the image geometrically equivalent, which simplifies the mathematical analysis<sup>4</sup>. We thus assume that the photoreceptors of the vision sensor are arranged on this unit sphere, each of them defining a viewing direction indicated by the unit vector  $d(\Psi, \Theta)$ , which is a function of azimuth  $\Psi$  and elevation  $\Theta$ . The 3D motion of this vision sensor or of its support-

<sup>3</sup>A unit radius allows to normalise the OF vectors on its surface and to express their amplitude directly in [rad/s].

<sup>4</sup>Ordinary cameras do not use spherical projection. However, if the FOV is not too wide, this approximation is reasonably close (Nelson and Aloimonos, 1989). A direct model for planar retinas can be found in Fermüller and Aloimonos (1997).

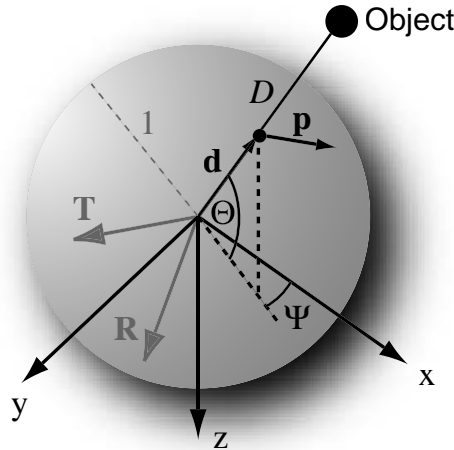


Figure 4.1: Spherical model of a visual sensor. A viewing direction indicated by the unit vector  $\mathbf{d}$ , which is a function of azimuth  $\Psi$  and elevation  $\Theta$ . The distance from an object in the direction  $\mathbf{d}(\Psi, \Theta)$  is denoted  $D(\Psi, \Theta)$ . The optic flow vectors  $\mathbf{p}(\Psi, \Theta)$  are always tangential to the sphere surface. The vectors  $\mathbf{T}$  and  $\mathbf{R}$  represent the translation and rotation of the visual sensor with respect to its environment.

ing body is characterised unambiguously by a translation vector  $\mathbf{T}$  and a rotation vector  $\mathbf{R}$  (describing the axis of rotation and its amplitude)<sup>5</sup>. When the vision sensor is moving in its environment, the motion field  $\mathbf{p}(\Psi, \Theta)$  is given by (Koenderink and van Doorn, 1987):

$$\mathbf{p}(\Psi, \Theta) = \left[ -\frac{\mathbf{T} - (\mathbf{T} \cdot \mathbf{d}(\Psi, \Theta)) \mathbf{d}(\Psi, \Theta)}{D(\Psi, \Theta)} \right] + [-\mathbf{R} \times \mathbf{d}(\Psi, \Theta)], \quad (4.1)$$

where  $D(\psi, \theta)$  is the distance between the sensor and the object seen in direction  $\mathbf{d}(\Psi, \Theta)$ . Although  $\mathbf{p}(\Psi, \Theta)$  is a 3D vector field, it is by construction tangential to the spherical sensor surface. Optic flow fields are thus generally represented by unfolding the spherical surface into a Mercator map (figure 4.2). Positions in the 2D space of such maps are also defined by the angles of azimuth  $\Psi$  and elevation  $\Theta$ .

Given a particular self-motion  $\mathbf{T}$  and  $\mathbf{R}$  and a specific repartition of distances  $D(\Psi, \Theta)$  from surrounding objects, equation (4.1) allows to reconstruct the resulting theoretical optic flow field. Beyond that, it formally supports a fact that was already suggested in chapter 2, i.e., the optic flow is a linear combination of the translatory and rotatory

<sup>5</sup>In the case of an aircraft,  $\mathbf{T}$  is a combination of thrust, slip, and lift, and  $\mathbf{R}$  a combination of roll, pitch, and yaw.

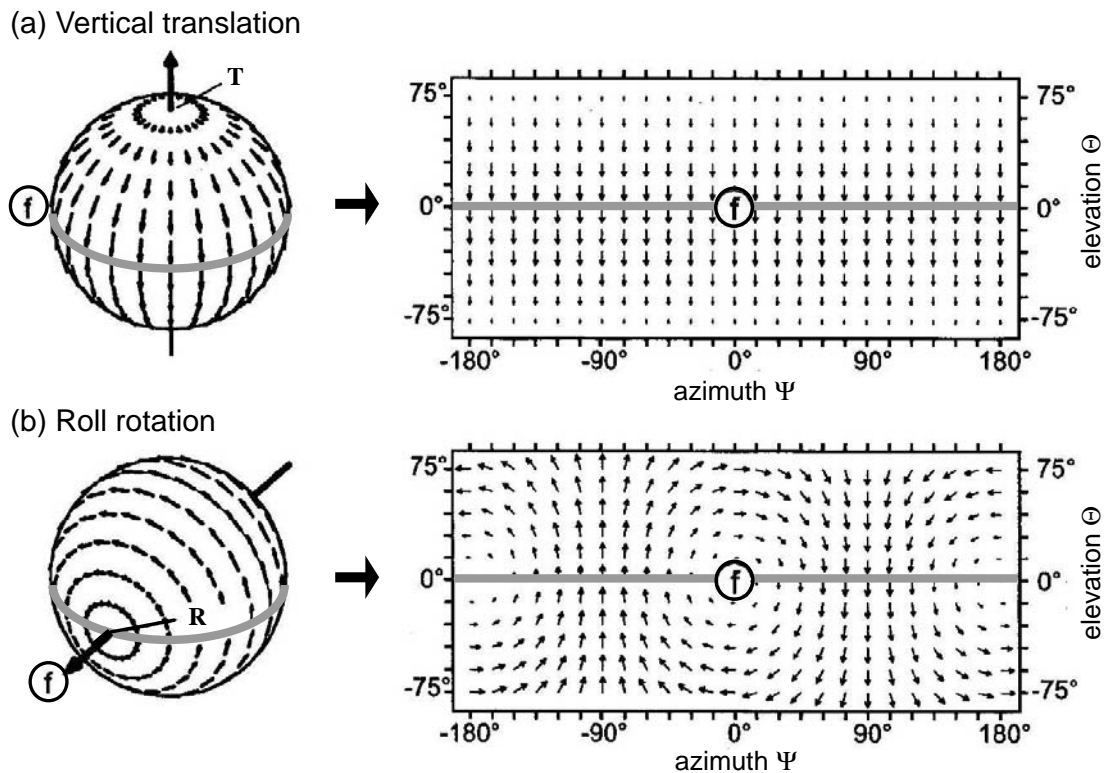


Figure 4.2: Optic flow fields due to (a) an upward translation and (b) a rotation around the roll axis. Projection of the 3D relative motion on spherical visual sensors (left) and development of the sphere surface into Mercator maps (right). The thick gray line highlights the equator of the spherical visual sensor, whereas the encircled “f” indicates the forward direction. See figure 4.1 for the definition of  $\Psi$  and  $\Theta$  angles. Adapted from Krapp and Hengstenberg (1996).

components<sup>6</sup> induced by the respective motion along  $T$  and around  $R$ . The first component, hereafter denoted *TransOF*, is due to translation and depends on the distance distribution, while the second component, *RotOF*, is produced by rotation and is totally independent of distances (figure 4.3).

From equation (4.1) we see that the *TransOF* amplitude is inversely proportional to distances  $D(\Psi, \Theta)$ . Therefore, if the translation is known and the rotation is null, it is in principle possible to estimate distances from surrounding objects. In free-maneuvring agents, however, the rotatory and translatory optic flow components are linearly superim-

<sup>6</sup>The local flow vectors in translatory OF fields are oriented along meridians connecting the focus of expansion (FOE, i.e., the direction point in which the translation is pointing at) with the focus of contraction (FOC, which is the opposite pole of the flow field). A general feature of the *RotOF* structure is that all local vectors are aligned along parallel circles centered around the axis of rotation.

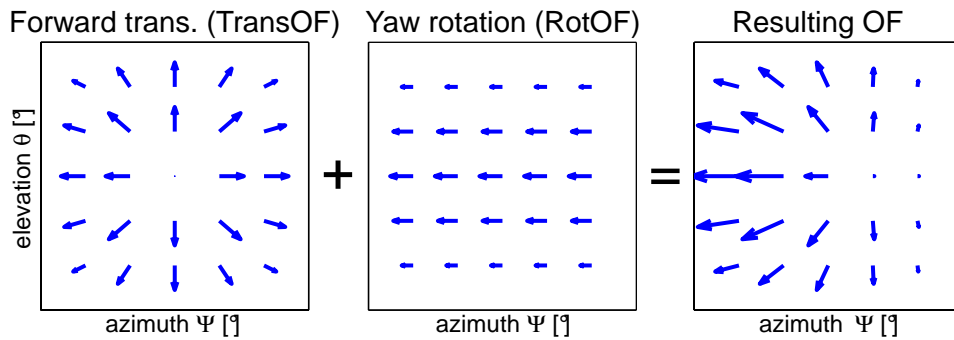


Figure 4.3: OF fields showing the effect of the superposition of TransOF and RotOF. The hypothetical camera is oriented toward a fronto-parallel plane. The first OF field is due to forward translation whereas the second one results from yaw rotation.

posed and may result in rather complex optic flow fields. It is quite common that RotOF overwhelms TransOF, thus making estimation of distance quite difficult. This is probably the reason why flies tend to fly straight and actively compensate for unwanted rotations (subsection 2.4.3). Another way of compensating for the spurious RotOF signals would consist in deducing it from the global flow field by measuring the current rotation with another sensory modality such as a gyroscope. Although this solution has not been shown to exist in insects, it will be adopted in our robots to compensate, in some sense, for the absence of active gaze stabilisation mechanisms.

### 4.1.3 Motion Parallax

A particular case of the general equation of optic flow (4.1) is often used in biology (Sobel, 1990; Horridge, 1977) and robotics (Franceschini et al., 1992; Sobey, 1994; Weber et al., 1997; Lichtensteiger and Eggenberger, 1999) to explain depth perception from optic flow. The so-called *motion parallax* refers to a planar situation where only pure translatory motion is assumed (figure 4.4). In this case, it is trivial<sup>7</sup> to express the optic flow amplitude  $p$  (also referred to as the *apparent angular velocity*) provoked by an object at distance  $D$ , seen under an azimuth  $\Psi$ :

<sup>7</sup>To derive the motion parallax equation (4.2) from the general optic flow equation (4.1), the rotatory component has first to be cancelled since no rotation occurs and then the translation vector  $\mathbf{T}$  needs to be expressed in the orthogonal basis formed by  $\mathbf{d}$  (the viewing direction) and  $\frac{\mathbf{p}}{\|\mathbf{p}\|}$  (the normalised optic flow vector).

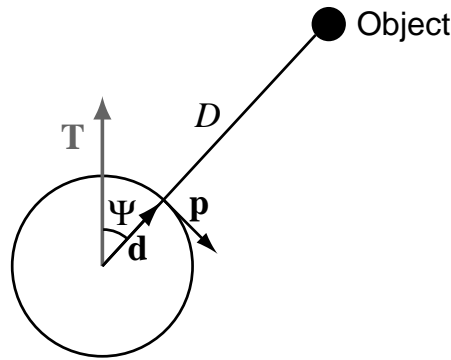


Figure 4.4: Motion parallax. The circle represents the retina of a moving observer and the symbols are defined in figure 4.1.

$$p(\Psi) = \frac{\|\mathbf{T}\|}{D(\Psi)} \sin \Psi, \text{ where } p = \|\mathbf{p}\|. \quad (4.2)$$

The formula has been first derived by Whiteside and Samuel (1970) in a brief paper about the blur zone that surrounds an aircraft flying at low altitude and high speed. If the translatory velocity and the optic flow amplitude are known, the distance from the object can thus be retrieved as follows:

$$D(\Psi) = \frac{\|\mathbf{T}\|}{p(\Psi)} \sin \Psi. \quad (4.3)$$

The motion parallax equation (4.2) is interesting in the sense that it gives a good intuition of how the optic flow varies on the retina.

## 4.2 Optic Flow Detection

Whereas the previous section gives an overview of ideal optic flow fields, here we are looking for an optic flow algorithm that will eventually lead to an implementation on the available hardware.

### 4.2.1 Issues with Elementary Motion Detectors

Within our bio-inspired approach, the most natural method for detecting optic flow would have been to use correlation-type EMDs<sup>8</sup> (sub-

<sup>8</sup>In fact, we started with a simple, discrete, correlation-type EMD implementation (after Iida and Lambrinos, 2000) and realised that it is indeed possible to have several of them implemented in the PIC microcontroller, working in real-time. However, the filter parameter tuning is tedious and, as expected, the EMD response is non-linear

section 2.3.2). However, beyond the fact that EMD models are still debated in biology and their spatial integration is not yet totally understood (subsection 2.3.2), the need for true image velocity estimates and insensitivity to contrast and spatial frequency of visual surroundings led us to turn away from this model.

It is often proposed (e.g., Harrison and Koch, 1999; Neumann and Bülthoff, 2002; Reiser and Dickinson, 2003) to linearly sum EMD signals over large receptive fields in order to smooth out the effect of non-linearities and other imprecisions. However, a linear spatial summation can produce good results only if a lot of detectable contrasts are present in the image, otherwise the spatial summation is highly dependent on the number of intensity changes (edges) capable of triggering an EMD signal. In our vertically striped experimental environments (section 3.4), the spatial summation of EMDs would be highly dependent on the number of viewed edges, which is itself strongly correlated with the distance from the walls. Even with random distribution of stripes, there is indeed more chance to see several stripes from far away than from close by. As a result, even if a triggered EMD will tend to increase its output with decreasing distances (as described in subsection 4.1.2), the number of active EMD in the field of view will simultaneously decrease. In such cases, the linear summation of EMDs will destroy the possibility of accurately estimating distances.

Although a linear spatial pooling scheme is suggested by the matched-filter model of the tangential cells (see figure 2.9) and has been used in several robotic projects (e.g., Neumann and Bülthoff, 2002; Franz and Chahl, 2002; Reiser and Dickinson, 2003), linear spatial integration of EMDs is not exactly what happens in the flies tangential neurons (subsection 2.3.3). Conversely, important non-linearities have been highlighted by several biologists (Hausen, 1982; Franceschini et al., 1989; Haag et al., 1992; Single et al., 1997), but are not yet totally understood.

### 4.2.2 Gradient-based Methods

An alternative class of optic flow computation has been developed within the computer-vision community (see Barron et al., 1994; Verri et al., 1992 for reviews), yielding output more independent of contrast or image structure.

---

with respect to image velocity and strongly depends on image contrast.

The standard approaches, the co-called *gradient-based methods* (Horn, 1986; Fennema and Thompson, 1979; Horn and Schunck, 1981; Nagel, 1982), assume that brightness (or intensity)  $I(n, m, t)$  of the image of a point in the scene does not change as the observer moves relative to it, i.e.:

$$\frac{dI(n, m, t)}{dt} = 0, \quad (4.4)$$

where  $n$  and  $m$  are the vertical and horizontal spatial coordinates in the image plane and  $t$  is the time. This equation can be rewritten as a Taylor series. Simple algorithms throw away the second order derivatives. In the limit as the time step tends to zero, we obtain the so-called *optic flow constraint equation*:

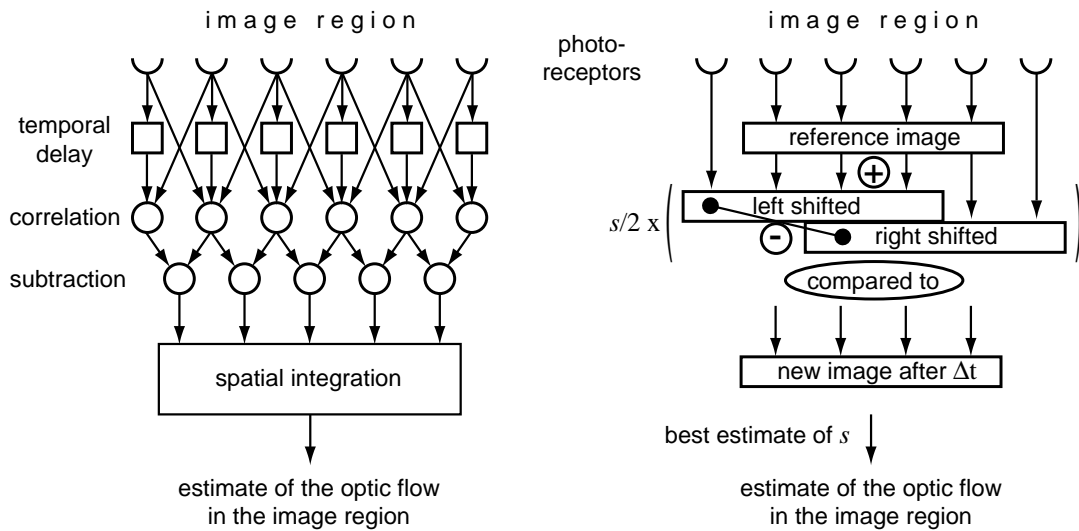
$$\frac{\partial I}{\partial n} \frac{dn}{dt} + \frac{\partial I}{\partial m} \frac{dm}{dt} + \frac{\partial I}{\partial t} = 0, \text{ with } \mathbf{p} = \left( \frac{dn}{dt}, \frac{dm}{dt} \right). \quad (4.5)$$

Since this optic flow constraint is a single linear equation in two unknowns, the calculation of the 2D optic flow vector  $\mathbf{p}$  is underdetermined. To solve this problem, one can introduce other constraints like, e.g., the *smoothness constraint* (Horn and Schunck, 1981; Nagel, 1982) or the assumption of *local constancy*<sup>9</sup>. Despite their differences, many of the gradient-based techniques can be viewed in terms of three stages of processing (Barron et al., 1994): (a) prefiltering or smoothing, (b) computation of spatiotemporal derivatives, and (c) integration of these measurements to produce a two-dimensional flow field, which often involves assumptions about the smoothness. Some of these stages often rely on iterative processes. As a result, the gradient-based schemes tend to be computationally intensive and very few of them are able to support real-time performance (Camus, 1995).

Srinivasan (1994) proposed an *image interpolation algorithm*<sup>10</sup> (I2A) in which the parameters of global motion in a given region of the image can be estimated by a single-stage, non-iterative process, which interpolates the position of a newly acquired image in relation to a set of older reference images. This technique is loosely related to a gradient-based method, but is superior to it in terms of its robust-

<sup>9</sup>The assumption that the flow does not change significantly in small neighbourhoods (local constancy of motion).

<sup>10</sup>This technique is quite close to the image registration idea proposed by Lucas and Kanade (1981). I2A has been further developed by Bab-Hadiashar et al. (1996), who quotes a similar methodology by Cafforio and Rocca (1976). A series of applications using this technique (in particular for self-motion computation) exists (Chahl and Srinivasan, 1996; Nagle and Srinivasan, 1996; Franz and Chahl, 2002; Chahl et al., 2004). The I2A abbreviation is due to Chahl et al. (2004).



(a) Spatial integration of EMDs over an image region      (b) I2A applied to an image region

Figure 4.5: EMD - I2A comparison (unidimensional case). (a) Spatial integration of several elementary motion detectors (EMDs) over an image region. See figure 2.6 for details about internal functioning of an EMD. (b) The simplified image interpolation algorithm (I2A) applied to an image region. Note that the addition and subtraction operators are pixel-wise. The symbol  $s$  denotes the image shift along the 1D array of photoreceptors. See subsection 4.2.3 for details about the I2A principle.

ness to noise. This is because, unlike the gradient scheme, which solves the optic flow constraint equation (4.5), the I2A incorporates an error-minimising strategy (see next subsection and Srinivasan, 1994, for further comparison with other methods).

Instead of spatially integrating local measurements, I2A estimates the global motion of a whole image region covering a wider FOV (see figure 4.5). Unlike spatially integrated EMDs, the I2A output has thus no dependency on image contrast, nor on spatial frequency, as long as some image gradient is present somewhere in the considered image region.

### 4.2.3 Simplified Image Interpolation Algorithm

To meet the constraints of our hardware, the I2A has been adapted to 1D images and limited to pure shifts (image expansion or other deformations are not taken into account in our simplified algorithm). The implemented algorithm works as follows (see also figure 4.5b). Let  $I(n)$  denote the grey level of the  $n^{\text{th}}$  pixel in the 1D image array (note that,



in this case,  $n$  is discrete and roughly corresponds to the azimuth  $\psi$ ). The algorithm computes the amplitude of the translation  $s$  between an image region (hereafter simply referred to as the “image”)  $I(n, t)$  captured at time  $t$ , called *reference image*, and a later image  $I(n, t + \Delta t)$  captured after a small period of time  $\Delta t$ . It assumes that, for small displacements of the image,  $I(n, t + \Delta t)$  can be approximated by  $\hat{I}(n, t + \Delta t)$ , which is a weighted linear combination of the reference image and of two shifted versions  $I(n \pm k, t)$  of that same image:

$$\hat{I}(n, t + \Delta t) = I(n, t) + s \frac{I(n - k, t) - I(n + k, t)}{2k}, \quad (4.6)$$

where  $k$  is a small reference shift in pixels. The image displacement  $s$  is then computed by minimizing the mean square error  $E$  between the estimated image  $\hat{I}(n, t + \Delta T)$  and the new image  $I(n, t + \Delta t)$  with respect to  $s$ :

$$E = \sum_n \left[ I(n, t + \Delta t) - \hat{I}(n, t + \Delta t) \right]^2, \quad (4.7)$$

$$\frac{dE}{ds} = 0 \Leftrightarrow s = 2k \frac{\sum_n [I(n, t + \Delta t) - I(n, t)] [I(n - k, t) - I(n + k, t)]}{\sum_n [I(n - k, t) - I(n + k, t)]^2}. \quad (4.8)$$

In our case, the shift amplitude  $k$  is set to 1 pixel and the delay  $\Delta t$  is such to ensure that the actual shift does not exceed  $\pm 1$  pixel.  $I(n \pm 1, t)$  are thus artificially generated by translating the reference image by one pixel to the left and to the right, respectively.

Note that in this restricted version of the I2A, the image velocity is assumed to be constant over the considered region. Therefore, if we want to measure non-constant optic flow fields, I2A must be applied to several subregions of the image where the optic flow can be considered as constant. In practice, the implemented algorithm is robust to small deviations from this assumption, but will of course be totally confused if opposite optic flow vectors occur in the same subregion.

In the following, the software (I2A) and hardware (a subpart of the 1D camera pixels) will be referred to as an optic flow detector (OFD). Such an OFD differs from an EMD in several respects (subsection 2.3.2). In general it has a wider FOV that can be adapted (by changing the optics and/or the number of pixels) to the expected structure of the flow field. In some sense, it participates in the process of spatial integration by relying on more than two photoreceptors. However, it should

always do so in a region of reasonably constant OF. In principle, it has no dependency on contrast or on spatial frequency of the image and its output presents a good linearity with respect to image velocity as long as the image shift remains in the limit of one pixel (or  $k$  pixels, in the general case of equation 4.8).

#### 4.2.4 Algorithm Assessment

In order to assess this algorithm with respect to situations that could be encountered in real-world conditions, we performed a series of measurements using artificially generated 1D images (figure 4.6) where we compared the I2A output signal  $s$  to the actual shift of the images. A set of high-resolution, sinusoidal, 1D gratings were generated and subsampled to produce 50-pixel-wide images with different shifts from -1 to +1 pixel with 0.1 steps. The first column of figure 4.6 shows sample images from the series of artificially generated images without perturbation (case A) and maximal perturbation (case B). The first line of each graph corresponds to the I2A reference image whereas the following ones are the shifted versions of the reference image. The second column of figure 4.6 displays the OF estimation produced by I2A versus actual image shift (black lines) and the error  $E$  (equation 4.7) between best estimate images and actual ones (gray lines). If I2A is perfect at estimating the true shift, the black line should correspond to the diagonal. The third column of figure 4.6 highlights the quality of the OF estimate (mean square error) with respect to the degree of perturbation (from case A to case B). In this column, large OF mean square error (MSE) indicates bad OF estimation.

A first issue concerns the sharpness of the image. In OF estimation, it is customary to preprocess images with a spatial low-pass filter in order to cancel out high-frequency content and reduce risk of aliasing effects. This holds for I2A too and figure 4.6a shows the bad quality of OF estimation with binary images (i.e., only totally black or white pixels). This result is expected because the spatial interpolation is based on a first-order numerical differentiation, which fails to provide a good estimate of the slope in presence of discontinuities (infinite slopes). It is therefore important to low-pass filter images such that edges spread over several adjacent pixels. A trade-off has to be found, however, between binary images and totally blurred ones where no gradient could be detected anymore. A handy way of low-pass filtering images at no computational cost is to slightly defocus the optics.

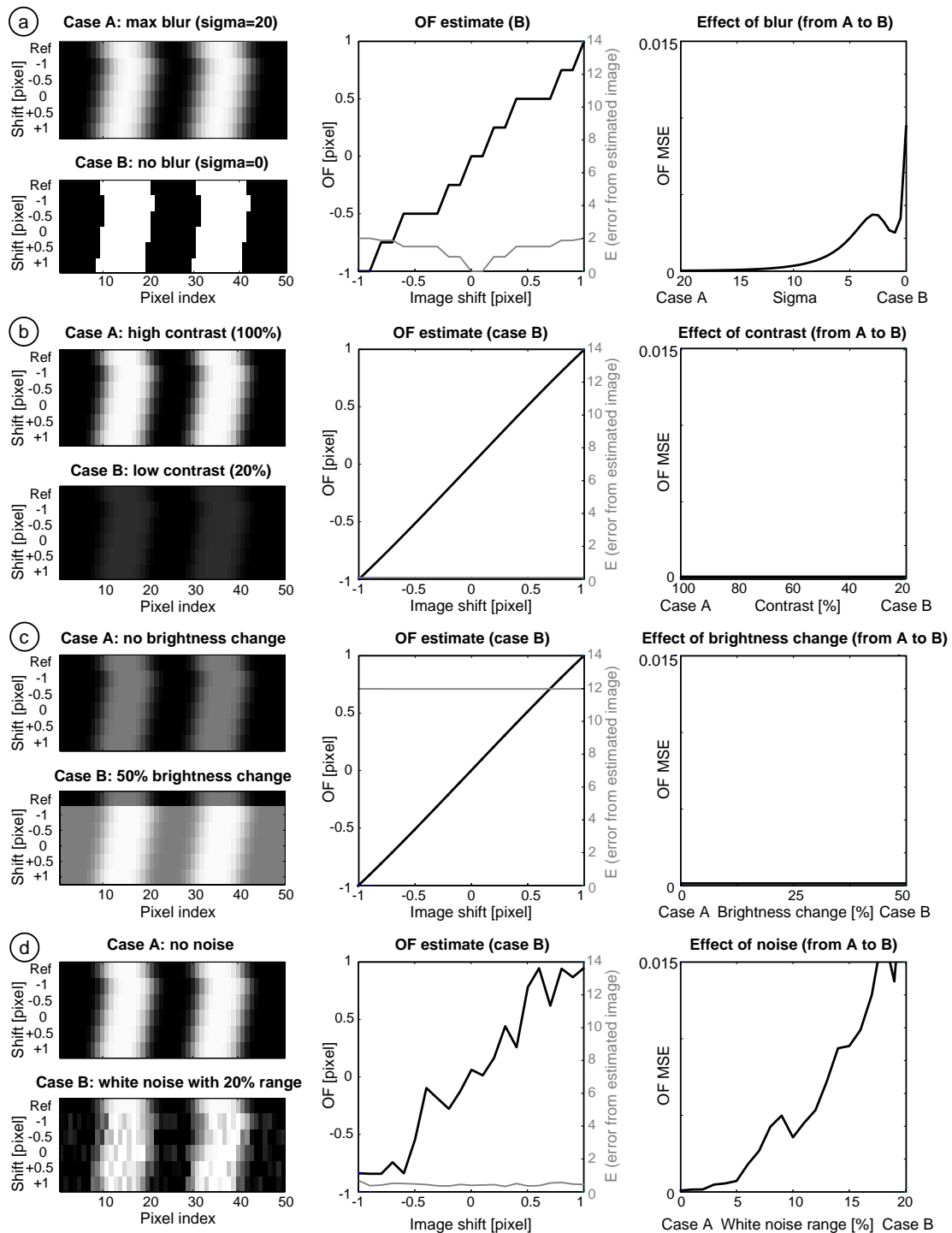


Figure 4.6: Study of perturbation effects on OF estimation. (a) Effect of Gaussian blur (sigma is the filter parameter). (b) Effect of contrast. (c) Effect of brightness change between reference image and new image. (d) Effect of noise. See text for details.

Low contrast<sup>11</sup> instead does not alter I2A estimates (figure 4.6b). As long as the contrast is not null, OF computation is performed reliably. This means that for a given image, there is almost no dependency on brightness settings of the camera, as long as the image gradient is not null. As a result, one can easily find a good exposition time setting and automatic brightness adjustment mechanisms could be avoided in most cases. Note that this analysis does not take noise into account and it is likely that noisy images will benefit from higher contrast in order to disambiguate real motion from spurious motion due to noise.

Another issue with simple cameras in artificially lit environments consists in the light flickering due to AC power sources, which could generate considerable change in brightness between the two successive image acquisitions of the I2A. Figure 4.6c shows what happens when the reference image is dark and the new image is up to 50% brighter. Here too, the algorithm performs very well, although, as could be expected, the error  $E$  is very large as compared to the other cases. This means that even if the best estimated image  $\hat{I}(n, t + \Delta t)$  is far from the actual new image because of the global difference in brightness, it is still the one that best matches the actual shift between  $I(n, t)$  and  $I(n, t + \Delta t)$ .

Another potential perturbation is the noise that can occur independently on each pixel (due to electrical noise within the vision chip or local optical perturbations). This has been implemented by the superposition of a white noise up to 20% in intensity to every pixel of the second image (figure 4.6d). The right-most graph shows that such kind of disturbance has minor effect up to 5%, while the center graph demonstrates the still qualitatively consistent although noisy OF estimate even with 20%. Although I2A is robust to a certain amount of noise, significant random perturbations like, e.g., those arising when part of the camera is suddenly dazzled because a lamp or other reflections enters the field of view could significantly affect its output. A temporal low-pass filter is thus implemented, which helps cancelling out such spurious data.

These results can be summarised as follows. The technique for estimating OF has no dependency on contrast as long as some image gradient can be detected. The camera should be slightly defocused to implement a spatial low-pass filter and it is not necessary to worry about flickering due to artificial lighting.

---

<sup>11</sup>Contrast is taken in the sense of the absolute difference between maximum and minimum intensity in the image.

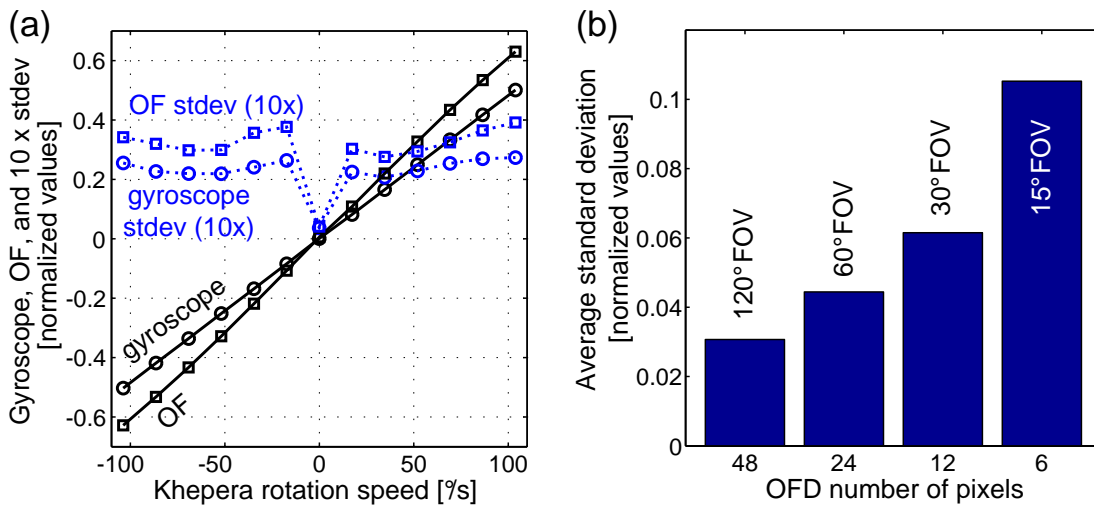


Figure 4.7: Experiment with a purely rotating Khepera in order to compare I2A output with gyroscopic data. Sensor values are normalised with respect to the entire range of a signed 8-bit integer ( $\pm 127$ ). (a) Gyroscope data (solid line with circles), related standard deviation of 1000 measurements for each rotation speed (dashed line with circles), and OF values estimated using 48 pixels (solid line with squares), related standard deviation (dashed line with squares). A value of 0.5 for the gyroscope correspond to  $100^\circ/\text{s}$ . Optic flow scale is arbitrary. (b) Average standard deviation of OF as a function of the FOV and corresponding pixel number. Adapted from Zufferey and Floreano (2004).

### 4.2.5 Implementation Issues

In order to build an OFD, equation (4.8) has been implemented in the embedded microcontroller, which grabs two successive images corresponding to  $I(n, t)$  and  $I(n, t + \Delta t)$  with a delay of a few milliseconds (typically 5-15ms) at the beginning of every sensory-motor cycle. Pixel intensities are encoded on 8 bits, whereas other variables containing the temporal and spatial differences are stored in 32-bit integers. For every pixel, equation (4.8) requires only 2 additions, 2 subtractions and 1 multiplication. Those operations are within the instruction set of the PIC microcontroller and can thus be executed very efficiently even with 32-bit integers. The only division of the equation occurs once per image region, at the end of the accumulation of the numerator and denominator. Since we are programming in C, this 32-bit division relies on a compiler built-in routine, which is executed in a reasonable amount of time since the entire computation for a region of 30 pixels is performed within 0.9ms. For comparison, a typical sensory-motor cycle lasts between 50 and 100ms.

In order to assess the OFD output in real-world conditions, the I2A algorithm was first implemented on the PIC of kevopic equipped with the frontal camera (subsection 3.2.2) and mounted on a Khepera. The Khepera was then placed in the 60x60cm arena (figure 3.12b) and programmed to rotate on the spot at different speeds. In this experiment, the output of the OFD can be directly compared to a gyroscope output. Figure 4.7a presents the results obtained from an OFD with an image region of 48 pixels roughly spanning a 120° FOV. Graph (a) illustrates the perfect linearity of the OF estimates with respect to the robot rotation speed. This linearity is in contrast with what could be expected from EMDs (see figure 2.7 for comparison). Even more striking is the similarity of the standard deviations between the gyroscope and OFD. This indicates that most of the noise, which is indeed very small, can be explained by mechanical vibrations of the Khepera (this is also why the standard deviation is almost null at 0°/s), and that the OFD is almost as good as the gyroscope at estimating rotational velocities. These measurements support our earlier suggestion of cancelling rotatory optic flow (RotOF) by simply subtracting a scaled version of the gyroscope value from the global OF. Note that rather than scaling the OFD output, one could simply adjust the delay  $\Delta t$  between the acquisition of the two successive images of I2A so to match the gyroscopic values in pure rotation.

To assess the effect of the FOV on the accuracy of an OFD output, we repeated the same experiment while varying the number of pixels. Note that for a given lens, the number of pixels is directly proportional to the FOV. However, this 120° lens (Marshall) is the one providing the lower angular resolution. Thus the results obtained in this experiments represent the worst case because it is obvious that the higher the resolution, the better the accuracy of the estimation. Figure 4.7b shows the average standard deviation of the OF measurements. The accuracy decreases reasonably until 12 pixels and 30° FOV. With only 6 pixels and 15°, the accuracy is 3 times worse than with 48 pixels. This trend may be explained by discretisation errors that tend to be better cancelled out with an higher number of pixels. Another factor is that a wider FOV provides richer images with more patterns allowing for a better match of the shifted images. At the limit, a too small FOV would sometimes have no contrast at all in the sampled image. When using such OFDs, a trade-off has thus to be found between a large enough FOV in order to ensure a good accuracy and a small enough FOV in order to better meet the assumption of local constancy of motion, when

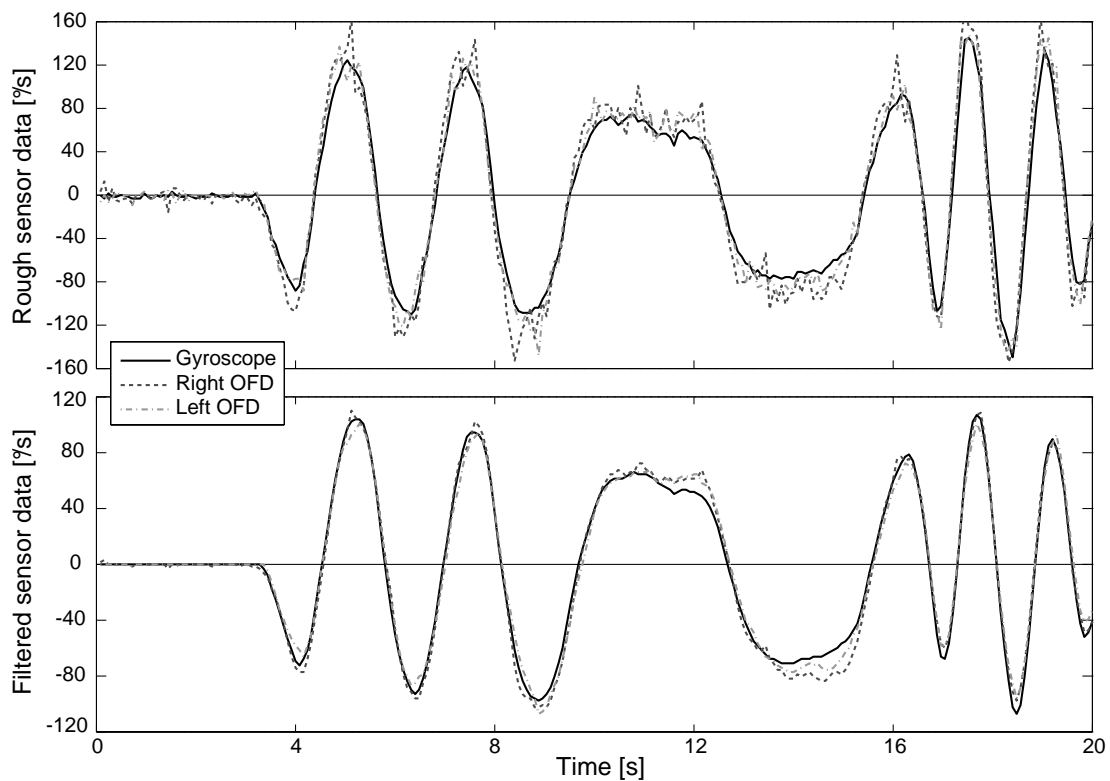


Figure 4.8: Match between gyroscope data and estimates of the right and left OFDs. The data have been recorded every 80ms while the F2 was held by hand in the experimental arena and randomly rotated around its yaw axis. The top graph displays the raw measurements, whereas the bottom graph shows their low-pass filtered version.  $100^\circ/\text{sec}$  is approximately the maximum rotation speed of the plane in flight. Adapted from Zufferey and Floreano (2005).

the robot is not undergoing only pure rotations.

To make sure that this approach still give good results in another configuration, we implemented two OFDs on the F2 airplane, one per camera (see figure 3.9 for the camera orientations). This time, we chose a FOV of  $40^\circ$  per OFD, which corresponds to 28 pixels with the EL-20 lens. The delay  $\Delta t$  was adjusted so to match the gyroscope output in pure rotation. The calibration provided an optimal  $\Delta t$  of 6.4ms. The airplane was then handled by hand and rotated about its yaw axis in its experimental arena (figure 3.12e). Figure 4.8 shows the data recorded during this operation and further demonstrate to the good match between rotations estimated by the two OFDs and by the gyroscope.

### 4.3 Summary

The first section of this chapter provides mathematical tools (equations 4.1 and 4.2) allowing to derive the amplitude and direction of optic flow given the self-motion of the agent and the geometry of the environment. Those tools will be used in chapter 5 both to decide how to orient the OFDs and to devise the control strategies using their outputs. Another important outcome of the formal description of optic flow is its linear separability into a translatory component (TransOF) and a rotatory component (RotOF). Only TransOF can provide useful information about distance from potential obstacles.

The second section presented the implementation of an optic flow detector (OFD) that fits the hardware constraints of the flying aircrafts while featuring linear response with respect to image velocity. Several of them can be implemented on a robot, each looking at different parts of the FOV (note that they could even have overlapping receptive fields) where the optic flow is assumed to be almost constant. Only the velocity component aligned with the 1D array of the camera can be estimated by such an OFD.



# Chapter 5

## Bio-inspired Navigation Control

When we try to build autonomous robots, they are almost literally puppets acting to illustrate our current myths about cognition.

---

I. Harvey (2000)

### Abstract

This chapter describes the development and assessment of robot controllers for autonomous indoor flight. It deals with higher level information processing than local optic-flow processing (chapter 4), such as spatial combination of optic-flow signals and integration of gyroscopic information in order to obtain efficient behaviours. The first section describes concrete cases of optic flow fields arising in typical phases of flight and discusses the position and orientation of the optic flow detectors. The second section presents the strategies for altitude control and steering control (course stabilisation and obstacle avoidance). The third section describes the assessment of the proposed control strategies, first on the Khepera then on the F2 airplane.

### 5.1 Characteristic Patterns of Optic Flow

A first integration issue consists in the choice of the position and orientation of the optic flow detectors (OFDs) on the robots. Equation (4.1) is employed to derive the typical optic flow (OF) patterns arising in particular flight situations. The chosen situations are those of (i) frontal approach toward a wall looking ahead to detect imminent collision and (ii)

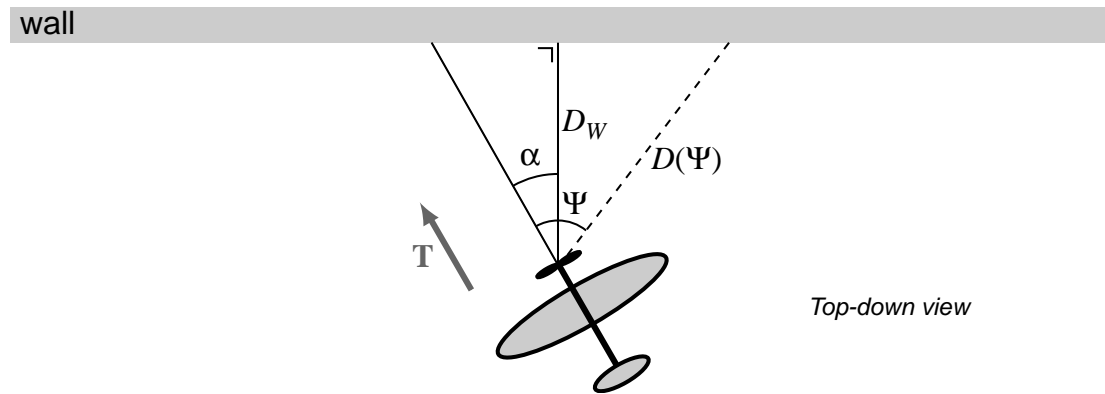


Figure 5.1: Frontal approach toward a flat surface (wall). The distance from the wall  $D_W$  is defined as the shortest distance (perpendicular to the wall surface). The approach angle  $\alpha$  is null when the translation  $\mathbf{T}$  is perpendicular to the wall.  $D(\Psi)$  represents the distance from the wall under a particular azimuth angle  $\Psi$ . Note that the drawing is a planar representation and in general  $D$  is a function not only of  $\Psi$ , but also of the elevation  $\Theta$ .

flying over a flat ground looking downward to perceive altitude. Since the rotatory optic flow (RotOF) does not contain any information about distances (subsection 4.1.2), this section focuses exclusively on translatory motion (the robots are assumed to move perfectly straight). The problem of cancelling RotOF will be tackled in subsection 5.2.3.

### 5.1.1 Frontal Approach

We consider the situation where the robot is approaching in straight and level flight an infinitely large wall at a given angle of approach  $\alpha$  (figure 5.1). The simplest case is the perpendicular approach to the wall ( $\alpha = 0^\circ$ ). Figure 5.2a displays the OF field arising in the frontal part of the FOV. This field is divergent, meaning that all OF vectors radiate from the focus of expansion (FOE). Note that the amplitude of the vectors are not proportional to their distance from the FOE (i.e.,  $\sqrt{\Psi^2 + \Theta^2}$ ). This would happen in the case where the distances  $D(\Psi, \Theta)$  from the obstacle are equidistant (i.e., a spherical obstacle centered at the visual sensor). With a flat surface, however, the distance increases as the elevation and azimuth angles depart from  $0^\circ$ . Since  $D(\Psi, \Theta)$  is in the denominator of the optic flow equation (4.1), this results in weaker OF amplitudes in the periphery. The locus of the viewing directions corresponding to maximum OF amplitudes is a circle going through  $\Psi = \Theta = \pm 45^\circ$  (de Talens and Ferretti, 1975).

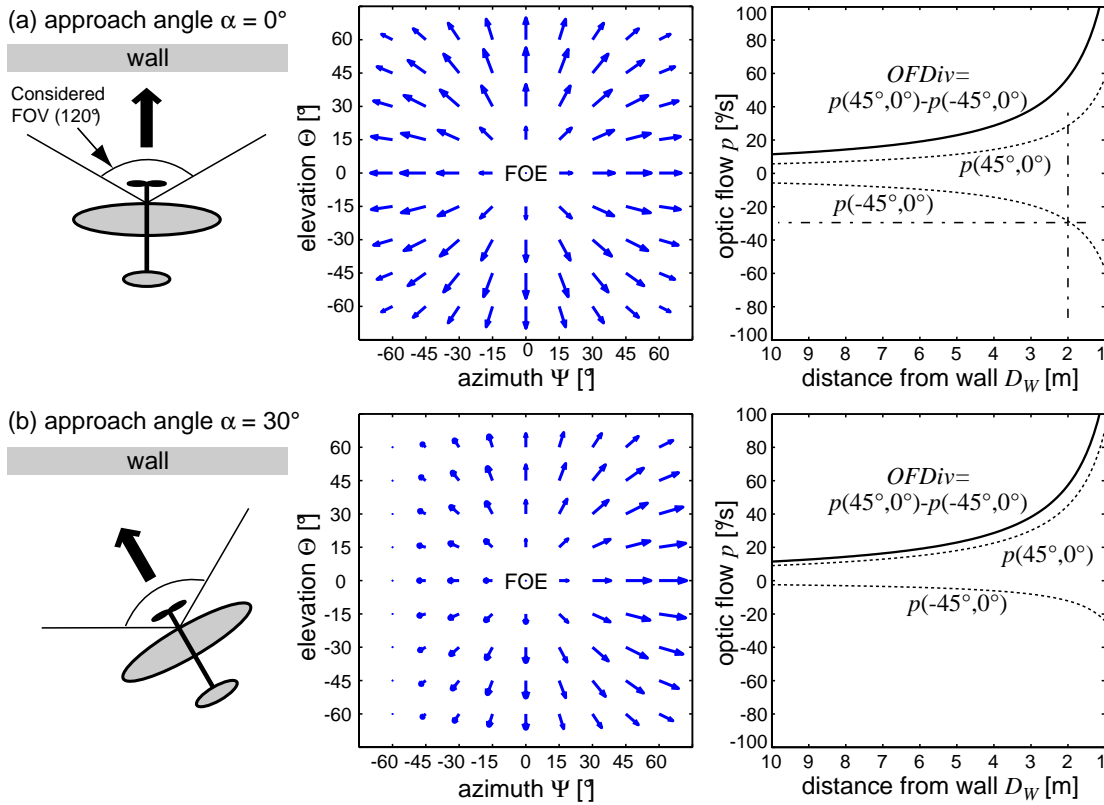


Figure 5.2: Ideal motion fields generated by forward motion at constant speed (2m/s). (a) Frontal approach toward a wall. (b) Approach at  $30^\circ$ . The first column depicts the the robot trajectory as well as the considered FOV. The second column shows the motion fields occurring in each situation. The third column shows the signed OF amplitudes  $p$  at  $\pm 45^\circ$  azimuth as a function of the distance from the wall  $D_W$ .

What happens when the distance from the obstacle  $D_W$  decreases over time as the plane is moving toward it? In figure 5.2a, third column, the signed<sup>1</sup> OF amplitude  $p$  at  $\Psi = \pm 45^\circ$  is plotted over time. Both curves are obviously symmetrical and the values are inversely proportional to  $D_W$ , as predicted by equation (4.1). Since those signals are asymptotic in  $D_W = 0$ , they constitute good cues for imminent collision warning. For instance, a simple threshold at  $p = \pm 30^\circ/s$  would be enough to trigger a warning 2m before collision (see vertical and hori-

<sup>1</sup>When considering the OF amplitude  $p = \|\mathbf{p}\|$ , we sometimes need to keep the information of the vector orientation, in particular when merging with gyroscope output. In this case, rightward and upward are positive, whereas leftward and downward are negative. The positive OF orientations have been chosen so to match the positive rotations around the yaw and pitch axes (see figure A.1 for the definition of the orientation of the body-fixed reference frame). Note that OF generated by positive self-rotation has a negative sign.

zontal dashed lines in figure 5.2a, right graph). According to equation (4.1), this distance fluctuates with the airplane velocity  $\|\mathbf{T}\|$ , but in a favorable manner. Since the optic-flow amplitude is proportional to the translational velocity ( $p \sim \|\mathbf{T}\|$ ), if the plane flies at 3m/s instead of 2m/s, the warning will be triggered earlier (at 3m instead of 2m before the wall), hence allowing more space for an avoidance action. In fact, in such a scheme with a fixed threshold on the OF, the value  $\frac{D_W}{\|\mathbf{T}\|}$  is constant. This value is nothing else than the time to contact (TTC, see subsection 2.3.3).

Based on these properties, it would be straightforward to place only one OFD directed in a region of maximum OF amplitude (e.g.,  $\Psi = 45^\circ$  and  $\Theta = 0^\circ$ ) to ensure a good signal to noise ratio and monitor when this value reaches a threshold. Note that whichever radially oriented location on the circle of radius  $\sqrt{\Psi^2 + \Theta^2} = 45^\circ$  could be chosen. However, in the real case of our airplane, the walls are not as wide as high (see figure 3.12e). As a result, OFDs oriented at non null elevation have a higher risk of looking at the ground or the ceiling. For this reason,  $\Psi = 45^\circ$  and  $\Theta = 0^\circ$  is the best choice.

What happens if the path direction is not perpendicular to the obstacle surface? Figure 5.2b depicts a situation where  $\alpha = 30^\circ$ . The OF amplitude on the left is weaker whereas the amplitude on the right is larger. In that case, the simplest solution is to sum (or average) left and right OF amplitudes, which produce the same curve as in the perpendicular case (compare the curves labelled *OFDiv*). This sum is proportional to the OF field divergence and is therefore denoted *OFDiv*. With this method<sup>2</sup> to detect imminent collision using a minimum number of OFDs, the *OFDiv* signal could be measured by summing two symmetrically oriented OFDs, both detecting OF along the equator.

Before adopting this method, it is interesting to consider how the OF amplitude behaves on the frontal part of the equator, when the plane is approaching the wall at different angles from  $0^\circ$  to  $90^\circ$  and what would be the consequences on *OFDiv*. This can be worked out using the motion parallax equation (4.2). The distance from the obstacle in each viewing direction (see figure 5.1 for the geometry and notations) is given by:

$$D(\Psi) = \frac{D_W}{\cos(\Psi + \alpha)}. \quad (5.1)$$

---

<sup>2</sup>This way of measuring OF divergence is reminiscent of the minimalist method proposed by Ancona and Poggio (1993), using Green's theorem (Poggio et al., 1991).

Then, using the motion parallax, the OF amplitude can be retrieved:

$$p(\Psi) = \frac{\|\mathbf{T}\|}{D_w} \sin \Psi \cdot \cos(\Psi + \alpha). \quad (5.2)$$

Building on this result, figure 5.3, left column, displays the OF amplitude in each azimuthal direction and for a set of approach angles ranging from  $0^\circ$  (perpendicular approach) to  $90^\circ$  (parallel to the wall). The second column plots the sum of the left and right sides of the first column graphs. This sum corresponds to *OFDiv* as if it would be computed for every possible azimuth in the frontal part of the equator. Up to  $\alpha = 30^\circ$ , the sum of OF remains maximum at  $|\Psi| = 45^\circ$ . For wider angles of approach, the peak shifts toward  $|\Psi| = 90^\circ$ .

Before drawing conclusions about optimal OFD viewing directions for estimating *OFDiv*, one should take into consideration the complexity of the avoidance manoeuvre, which depends essentially on the approach angle. When arriving perpendicularly to the wall, the airplane must perform at least a  $90^\circ$  turn. Instead, when following an oblique course (e.g.,  $\alpha = 45^\circ$ ), a  $45^\circ$  turn in the correct direction is enough to avoid a collision, and so on until  $\alpha = 90^\circ$  where no avoidance action is required at all. In a situation with two OF measurements at  $\Psi = \pm 45^\circ$ , the *OFDiv* signal (figure 5.3, right column) will be 100% when the plane is approaching perpendicularly and will decrease to 70% at  $45^\circ$ , and to 50% at  $90^\circ$  (where no action is required). As a result, the imminent collision detector will trigger 30% closer to the wall when the approach angle is  $45^\circ$ . The plane could also fly along the wall ( $\alpha = 90^\circ$ ) without any warning at a distance 50% closer to the wall than if it would have flown on a perpendicular trajectory. Therefore, this scheme for detecting approaching objects and imminent collision is particularly interesting for obstacle avoidance, because it automatically adapts the occurrence of the warning to the angle of approach and the corresponding complexity of the avoidance manoeuvre.

A similarly interesting trend of the *OFDiv* signal computed as a sum of left and right OF amplitudes happens in the case of approaching a corner (figure 5.4). Here the minimal avoidance action is even greater than in the worst situation with a simple wall because the plane should turn by more than  $90^\circ$  (e.g.,  $135^\circ$  when approaching on the bisectrix). Fortunately, the *OFDiv* signal is significantly higher in that case because the average distances from walls are smaller (compare *OFDiv* curve in figure 5.4 and 5.2).

In summary, two OFDs are theoretically sufficient for detecting im-

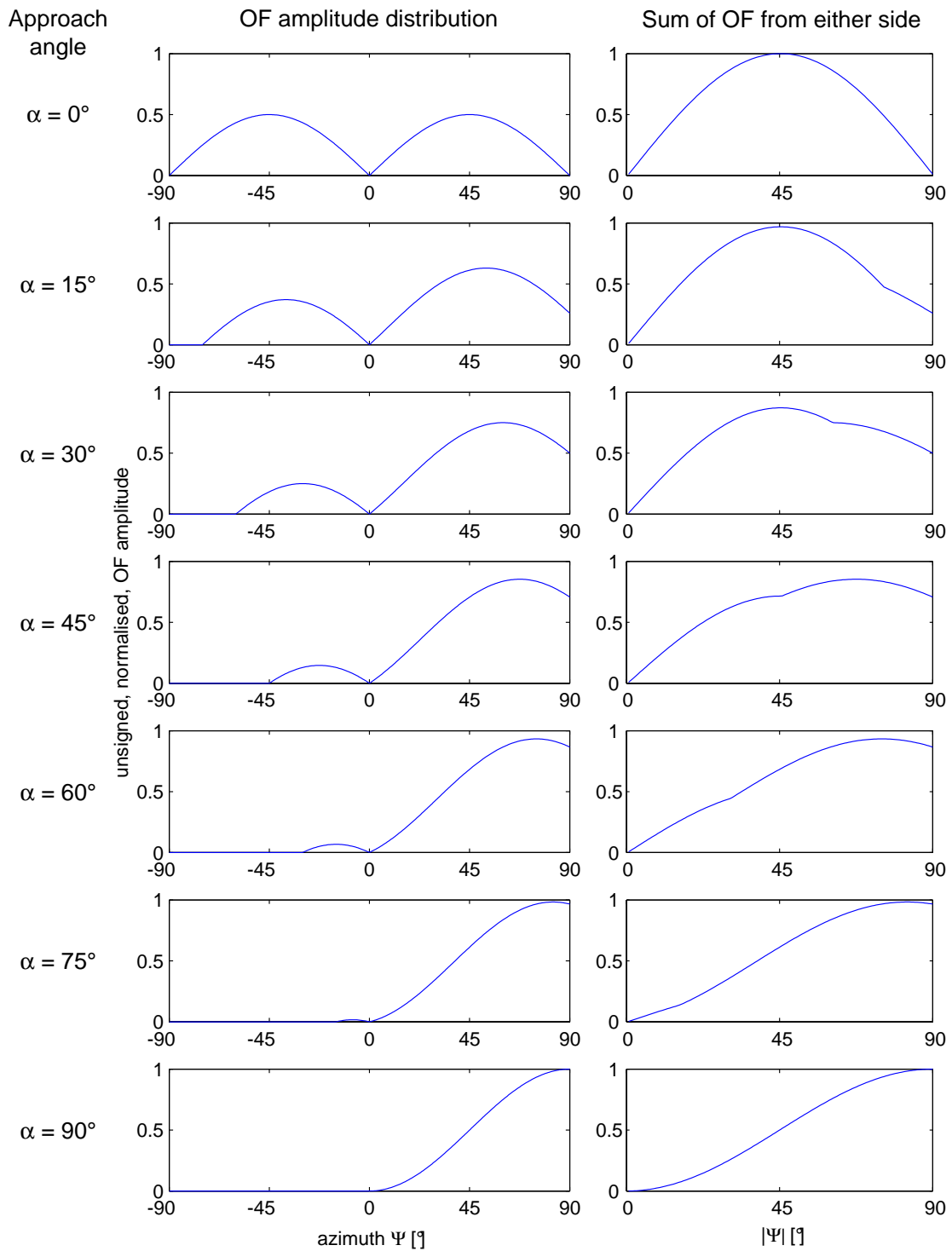


Figure 5.3: Series of graphs showing the repartition of the unsigned, normalised, OF amplitudes on the equator of the vision sensor (i.e., where  $\Theta = 0^\circ$ ) in the case of a frontal approach toward a flat surface at different approach angles  $\alpha$ . The second column represents the symmetrical sum of left and right OF amplitudes, as if the graphs on the left were folded vertically at  $\Psi = 0^\circ$  and the OF values for every  $|\Psi|$  were summed together.

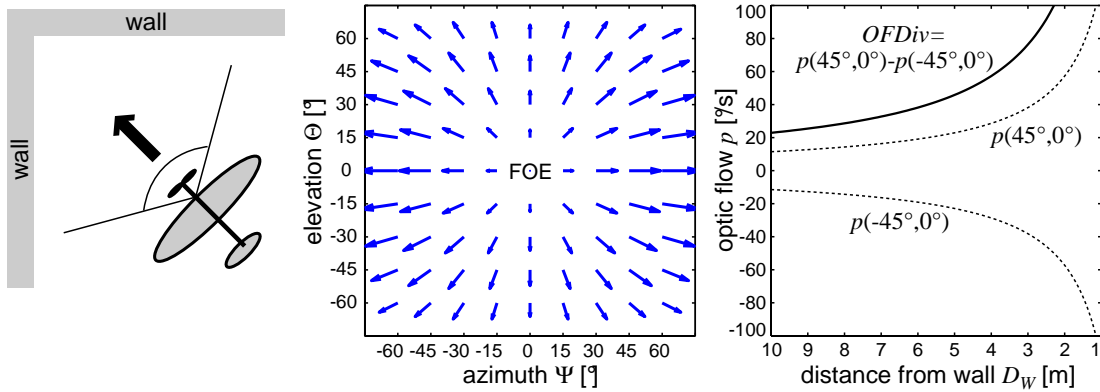


Figure 5.4: Same as figure 5.2, but for the case of an approach toward a wall corner.

minent collisions. The best way of implementing them on the robot is to orient their viewing directions such as  $\Psi = \pm 45^\circ$  and  $\Theta = 0^\circ$  and place them horizontally in order to detect OF along the equator. Summing their output would create an  $OFDiv$  signal that can be used with a simple threshold to detect impending collision. A further interesting property of this signal is that it reaches the same value at slightly different distances from obstacles, and the way this distance varies is adapted (i) to the complexity of the minimal avoidance action (the required turning angle), and (ii) to the velocity of flight.

### 5.1.2 Flying Over

The second situation of interest is the flight over a flat ground (figure 5.5). The typical OF pattern occurring in the bottom part of the FOV is simpler than in the previous situations. As seen in figure 5.6, all OF vectors are collinear. Following equation (4.1), their amplitude is inversely proportional to the distances from the ground ( $p \sim \frac{1}{D(\Psi, \Theta)}$ ). The maximum OF amplitude in case of level flight is located at  $\Theta = -90^\circ$  and  $\Psi = 0^\circ$ . Therefore, a single OFD oriented in this direction, i.e., vertically downward, could be a good solution to estimate altitude because its output is proportional to  $\frac{1}{D_A}$ .

Now let us restrict the problem to 2D (as shown in figure 5.5), and analyse what happens to the OF field along  $\Psi = 0^\circ$  when the airplane makes pitch adjustments in order to correct its altitude. As before, the motion parallax equation (together with the distance from ground at each elevation angle) allows to gain better insight into the problem (see figure 5.5 for the geometry and notations):

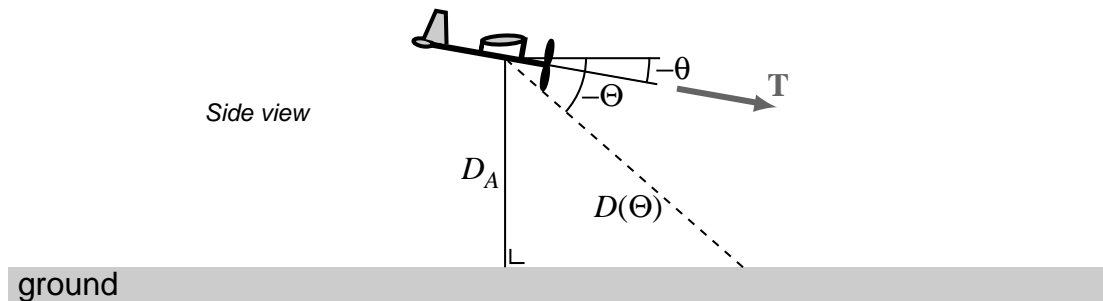


Figure 5.5: Flying over a flat surface (ground). The distance from the ground  $D_A$  (altitude) is defined as the shortest distance (perpendicular to the ground surface). The pitch angle  $\theta$  is null when  $\mathbf{T}$  is parallel to the ground.  $D(\Theta)$  represents the distance from the ground at a particular elevation angle  $\Theta$  in the visual sensor reference frame. Note that the drawing is a 2D representation and in general  $D$  is a function not only of  $\Theta$ , but also of the azimuth  $\Psi$ .

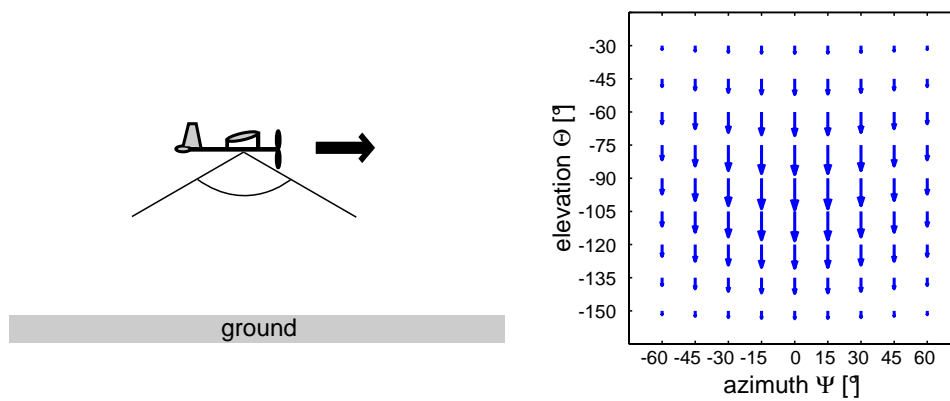


Figure 5.6: Ideal motion field generated by forward motion at constant speed (2m/s) over a flat ground. The considered FOV in this case is oriented vertically toward the ground.



$$D(\Theta) = \frac{D_A}{-\sin(\Theta + \theta)} \Rightarrow p(\Theta) = \frac{\|\mathbf{T}\|}{D_A} \sin \Theta \cdot \sin(\Theta + \theta). \quad (5.3)$$

Equation (5.3) gives the OF amplitude as a function of the elevation for different cases of negative pitch angles (figure 5.7). Of course, the situation is symmetrical for positive pitch angles. Based on this result, the plots in figure 5.7 reveal that the location of the maximum OF is  $\Theta = -90^\circ$  plus half the pitch angle. For example, if  $\theta = -30^\circ$ , the peak is located at  $\Theta = -90 - 30/2 = -75^\circ$  (see vertical dashed line in the third graph). This property can be derived mathematically from equation (5.3):

$$\frac{dp}{d\Theta} = \frac{\|\mathbf{T}\|}{D_A} \sin(2\Theta + \theta) \quad \text{and} \quad \frac{dp}{d\Theta} = 0 \iff \Theta_{max} = \frac{\theta + k\pi}{2}. \quad (5.4)$$

The peak amplitude weakens only slightly when the pitch angle departs from  $0^\circ$ . Therefore, only one OFD pointing vertically downward is likely to provide sufficient information to control altitude, especially with an airplane that will hardly exceed  $\pm 10^\circ$  of pitch angle.

However, in situations where the pitch angle is not limited to small values, several OFDs could be implemented, each oriented at different elevation angles. Then only the OFD providing maximum output signal should be considered. This would allow to automatically follow the peak location (whose value is directly related to the altitude) and, in some sense, track the minimum distance to the ground. Although not used nor tested in this thesis, this could also give an estimation of the current pitch angle of the plane and thus provide a vertical reference (see also subsection 2.4.1).

## 5.2 Information Processing and Control

This section describes how to combine the biological models presented in chapter 2 and the insights into typical OF patterns gained in the previous section in order to devise efficient control strategies relying on sensory information from several OFDs and a gyroscope. Since attitude control (ATC) is not required<sup>3</sup>, control strategies are proposed only for course stabilisation (CS), obstacle avoidance (OA) and altitude control (ALC). Theoretically, this minimal set of control strategies is sufficient to drive a flying robot in enclosed environments. In the next section,

<sup>3</sup>For a discussion about passive attitude stability of the F2, see subsection 3.1.3.

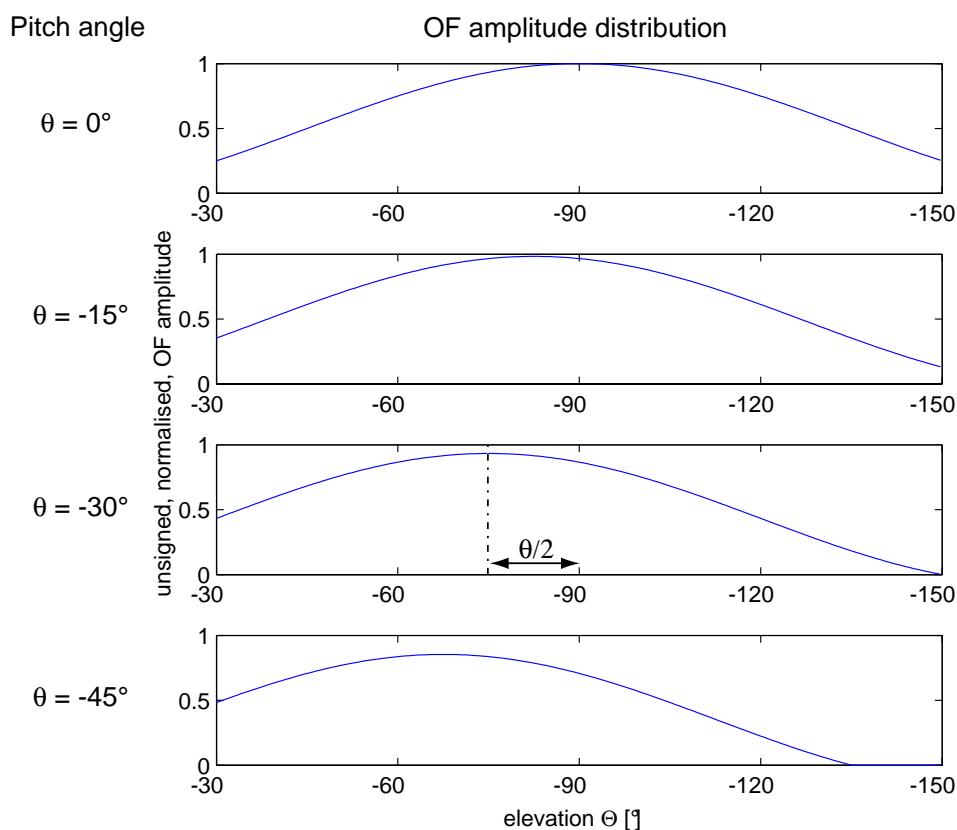


Figure 5.7: Series of graphs showing the repartition of the unsigned, normalised, OF amplitudes in the longitudinal direction (i.e.,  $\Psi = 0^\circ$ ) in the case of a flight over a flat surface at different pitch angles  $\theta$ .

assessment of OA and ALC are done with the Khepera, whereas autonomous steering (CS and OA) is demonstrated with the 30-gram airplane.

### 5.2.1 Steering Control (CS & OA)

The steering strategy (CS and OA) is largely inspired by a recent study by Tammero and Dickinson (2002b) on the behaviour of free-flying fruitflies (subsection 2.4.3). They showed that

- OF divergence experienced during straight flight sequences is responsible for triggering saccades,
- the direction of the saccades (left or right) is opposite to the side experiencing larger OF, and
- during saccades no visual feedback seems to be used.

The proposed steering strategy can thus be divided into two states: (i) maintain straight course (CS) and (ii) turn as quickly as possible as soon as an imminent collision is detected (OA).

### **Course Stabilisation (CS)**

Maintaining a straight course is important in two respects. On the one hand, it spares energy in flight because when a plane banks, it must produce additional lift in order to compensate for the centrifugal force. On the other hand, it provides better conditions for estimating OF because the airplane is in level flight and frontal OFDs will see only the contrasted walls and not the ceiling and floor.

In subsection 2.4.2, it was said that flying insects are believed to implement CS using both visual and vestibular cues. In order to achieve straight course with our artificial systems, we propose to rely exclusively on gyroscopic data. It is likely that the artificial gyroscope has higher accuracy than the halteres' system, especially at low rotation rates. Moreover, decoupling the sensory modalities by attributing the gyroscope to CS and vision to OA allows to simplify the control structure.

With an airplane, CS can thus be easily implemented by means of a proportional feedback loop connecting the gyroscope to the rudder servomotor. Note that, unlike the plane, the Khepera does not need a gyroscope for moving straight since its wheel speeds are regulated and almost no slipping occurs between the wheels and the ground. Issuing the same speed command to both wheels will force the robot to move straight.

### **Obstacle Avoidance (OA)**

A saccade (quick turning action) allows to avoid collisions. To detect imminent collisions, we propose to rely on the spatio-temporal integration of motion (STIM) model (subsection 2.3.3), which spatially and temporally integrates optic flow from the left and right eyes. Note that according to Tammero and Dickinson (2002a), the STIM model remains the one that best explains the landing and collision-avoidance responses in their experiments. Considering this model from an engineering viewpoint, imminent collision can be detected during straight motion using the  $OF_{Div}$  signal obtained by summing left and right OF amplitudes measured at  $\pm 45^\circ$  azimuth, (subsection 5.1.1). Therefore,

two OFDs must be mounted horizontally and oriented at  $45^\circ$  on both sides of the forward direction. Let us denote  $LOFD$  the output signal of the left detector and  $ROFD$  that of the right one,  $OFDiv$  is thus obtained as follows:

$$OFDiv = ROFD + (-LOFD). \quad (5.5)$$

Note that OFD output signals are signed OF amplitudes that are positive for rightward motion. In order to prevent noisy transient OFD signals (that may occur long before an actual imminent collision) to trigger a saccade, the  $OFDiv$  signal is low-pass filtered. Figure 5.8 outlines the comparison between the fly model and the system proposed for the robots. Note that a leaky integrator (equivalent to a low-pass filter) is also present in the fly model and accounts for the fact that weak motion stimuli do not elicit any response (Borst, 1990).<sup>4</sup>

As pointed out in subsection 5.1.1, the output signal  $OFDiv$  will reach the threshold in a way that depends on the speed, the angle of approach and the geometry of the obstacle. For instance, the higher the approach speed, the further from obstacle the trigger will occur.

### Choosing the Saccade Direction

As seen in chapter 4, close objects generate larger translatory optic flow. The left-right asymmetry OFD outputs prior to each saccade is thus used in order to decide the direction of the saccade. The same strategy seems to be used by flies to decide whether to turn left or right (Tammero and Dickinson, 2002b). A new signal is thus defined, which measures the difference between left and right absolute OF values:

$$OFDiff = |ROFD| - |LOFD|. \quad (5.6)$$

A closer obstacle on the right results in a positive  $OFDiff$ , whereas a closer obstacle on the left produces a negative  $OFDiff$ .

Finally, figure 5.9 shows the overall signal flow diagram for saccade initiation and direction selection. The graph already integrates the RotOF cancellation mechanism that will be described in subsection 5.2.3.

---

<sup>4</sup>However, the time constant of the low-pass filter could not be precisely determined.

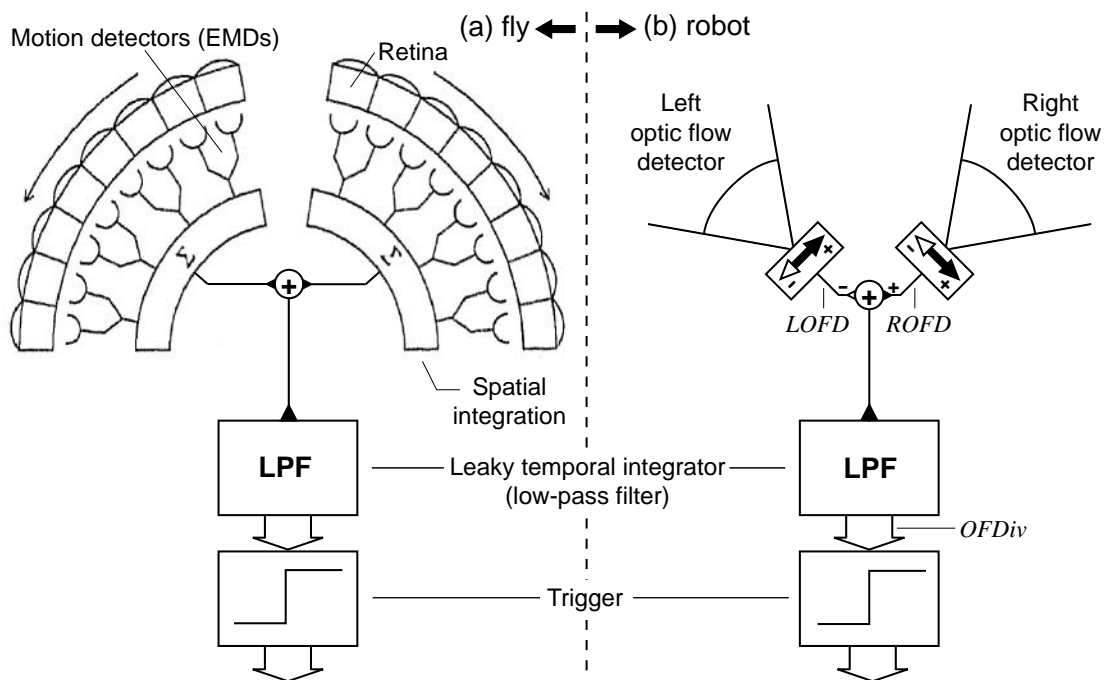


Figure 5.8: The STIM model (on the left, adapted from Borst and Bahde, 1988) as compared to the system proposed for our robots (on the right). (a) The output of motion detectors (EMDs) sensitive to front-to-back motion are spatially pooled from each side. The resulting signal is then fed into a leaky temporal integrator (functionally equivalent to a low-pass filter). When the temporal integrator reaches a threshold, a preprogrammed motor sequence can be performed, either to extend legs or to trigger a saccade (see subsection 2.3.3 for further discussion). (b) The system proposed for imminent collision detection in our robots is very similar. The spatial pooling of EMDs on left and right regions of the field of view are replaced by two OFDs.

### 5.2.2 Altitude Control (ALC)

As suggested in subsection 2.4.4, altitude can be controlled by holding ventral optic flow constant. This idea is based on experiments with honey-bees that seem to use this kind of mechanism for tasks like grazing landing and control of flight speed. A more detailed analysis of the typical patterns of ventral optic flow has been presented in subsection 5.1.2, highlighting the loci of maximum OF amplitude when the pitch angle varies. As long as these variations are kept small (typically  $\pm 10^\circ$ ), it is reasonable to use only one ventral OFD. As proposed in subsection 5.1.2, if larger pitch angles can occur, it is worth to track the peak OF value. In this case, several OFDs pointing at different elevations must be implemented and only the OFD producing the maximum output (winner-take-all) is taken into account in the control loop.

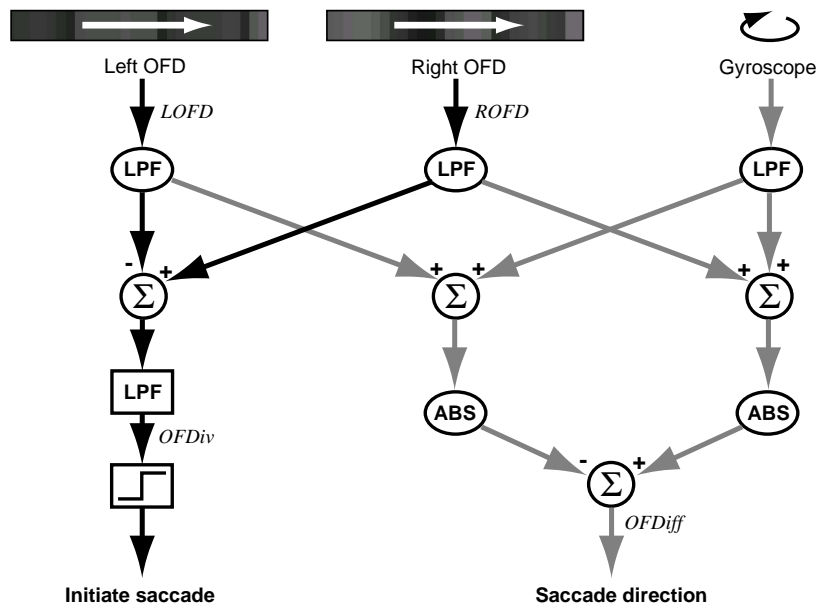


Figure 5.9: Signal flow diagram for saccade initiation (obstacle avoidance) based on horizontal OF divergence and rotation rate as detected by the yaw gyroscope. Arrows on top of the diagram indicate the positive directions of OFDs and gyroscope (note that rightward self-rotation produces negative or leftward OF). LPF stands for low-pass filter and ABS is the absolute value operator. Signal from OFDs and gyroscope are first low-pass filtered to cancel out high-frequency noise (see figure 4.8). Below this first-stage filtering, one can recognise, on the left (black arrows), the STIM model responsible for saccade initiation and, on the right (gray arrows), the pathway responsible for deciding whether to turn left or right.

The type of control loop linking ventral OF amplitude to the pitch angle should be proportional and derivative in order to counterbalance the integrative effect of the pitch angle  $\theta$  on the variation of the altitude, i.e.,  $\frac{dD_A}{dt} \sim \theta$  (see figure 5.5).

### 5.2.3 Rotatory Optic Flow Cancellation

The above control strategies are likely to fail whenever a rotational movement occurs. For instance, a superimposed pitching rotatory OF signal (RotOF, see subsection 4.1.2) would have a dramatic impact on the altitude control loop. This will happen whenever the control system itself is acting on the elevator to change the pitch angle. The same holds for the horizontal OFDs used for obstacle avoidance. Every modification of trajectory by the course stabilisation mechanism or, more prosaically, every air turbulence will produce some RotOF.

As seen in chapter 4, these RotOF components do not contain any information about surrounding distances, and for all kinds of tasks related to distance, a pure translatory OF field is desirable (Srinivasan et al., 1996). This holds for the robots just as it does for the fly, which is known to compensate with its head for rotations detected by its halteres (subsection 2.4.2). Since our airplane cannot afford additional actuators to move its cameras, we propose another means of cancelling RotOF, based on the same sensory modality used by flies.

It is in principle possible to deduce RotOF from the global flow field by simple vector subtraction, because the global OF is a linear combination of translatory and rotatory components (subsection 4.1.2). To do so, it is necessary to know the rotation, which can be measured with another sensory modality such as a gyroscope. In our case the situation is quite trivial because the OFDs are unidimensional and a gyroscope is mounted with its axis oriented perpendicular to the pixel array and the viewing direction (see subsection 3.2.2). This arrangement reduces the correction operation to a scalar subtraction. Of course a simple subtraction can be used only if the optic flow detection is linearly dependent on the rotation speed (which is not the case of EMDs). Subsection 4.2.5 further supports this method of RotOF deduction by demonstrating the good match between OFD signals and gyroscope output in pure rotation.

Note that,  $OFDiv$  as computed in equation (5.5) is not sensitive to yaw rotation because the rotatory component is equally detected by the two OFDs, whose outputs are subtracted.<sup>5</sup> Unlike  $OFDiv$ ,  $OFDiff$  does suffer from RotOF and must be corrected with the gyroscope signal (figure 5.9). The same holds for any OF signals used in altitude control.

## 5.3 Experiments and Results

This section describes real-world test of the proposed control strategies. First a preliminary test of altitude control (ALC) is carried out on the Khepera robot by transposing the problem into a wall following situation. Saccadic obstacle avoidance (OA) is then implemented on the wheeled robot as a preliminary step toward the final experiment consisting in autonomous steering (CS and OA) of the 30-gram airplane.

---

<sup>5</sup>This property is also pointed out by Ancona and Poggio (1993). The method for computing flow divergence is independent on the location of the focus of expansion. This means that the measured divergence remains unaltered, even if the FOE shifts due to rotation.

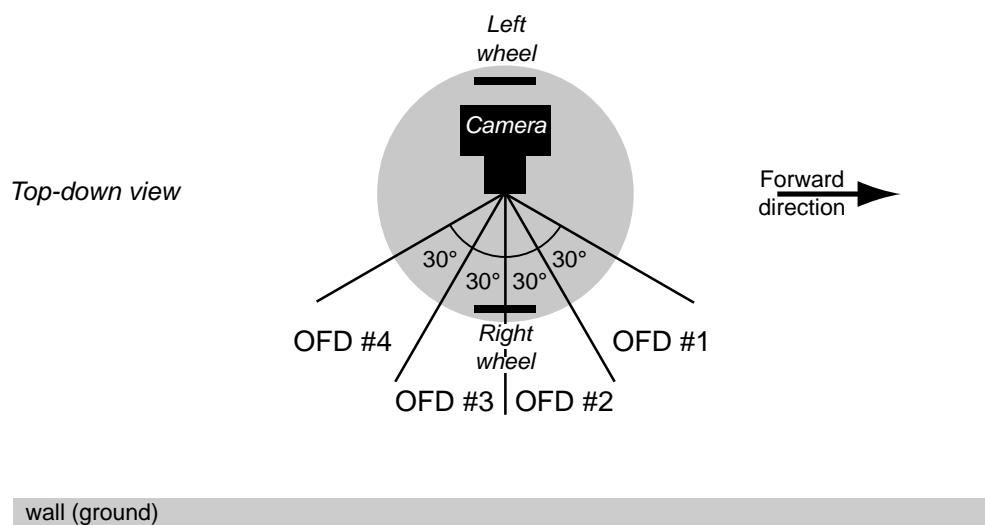


Figure 5.10: Outline of the Khepera equipped with the wide FOV lateral camera (see also figure 3.9a) for the wall-following experiment. Four OFDs are implemented, each using a subpart of the pixels.

### 5.3.1 Altitude Control as Wall Following on Wheels

In order to assess the altitude control mechanism suggested in subsection 5.2.2, we implemented it as a wall following mechanism in the Khepera (kevic) with the camera oriented laterally (figure 5.10). In this situation, the distance from the wall would correspond to the altitude of the aircraft. Four OFDs are implemented each using a subpart of the pixels of the single onboard 1D camera.

A proportional-derivative controller tries to hold the OF amplitude constant by acting on the differential speed between the left and right wheels, which roughly corresponds to an elevator deflection command on an airplane. As described in the previous subsection, yaw gyroscope signal is used to remove the spurious RotOF occurring when the differential speed is not null. The OFD value used by the controller is always the one producing the highest output among the four OFDs. In practice, only the two central OFDs are often in use, but it can happen that the external ones are used when the Khepera takes very steep angles with respect to the wall.

Several tests have been performed with a 120cm-long wall (figure 5.11). Although the robot does not always keep the same distance from the wall, these tests show that such a simple control strategy based on optic flow could produce reliable altitude control. Note that this would not have been possible without RotOF correction. This control



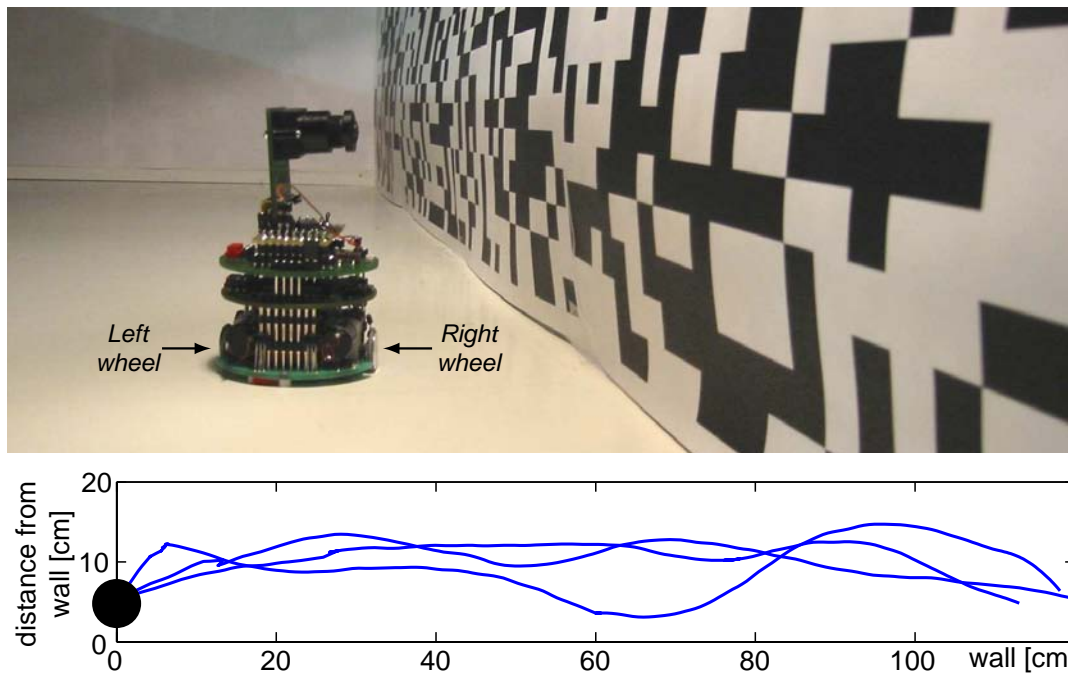


Figure 5.11: Altitude control (implemented as wall following) with the Khepera. Top: 120cm long setup and the Khepera with the lateral camera. Bottom: Wall following results (3 trials). The black circle indicates the robot's initial position. Trajectories are reconstructed from wheel encoders. Adapted from Zufferey and Floreano (2005).

strategy relies on no other sensors than vision and gyroscope. It is therefore a good candidate for an ultra-light flying aircraft. However, this experiment remains a proof of concept and no further tests have been made on aerial robots at this stage. A few problems are likely to arise with an aerial robot. Among others, the fluctuation of forward velocity whenever the pitch angle is modified could provoke instability in the control loop. See section 6.4 for further discussion about how this altitude control strategy could be implemented on an airplane and what are the main concerns.

### 5.3.2 Steering Control on Wheels

The steering control proposed in subsection 5.2.1 (without course stabilisation that is not required on wheels) was tested in a square arena (figure 3.12b) with the Khepera equipped with the frontal camera (figure 5.12). Two OFDs with a FOV of  $30^\circ$  are implemented using 50% of the available pixels. The  $OFDiv$  signal is computed by subtracting the output of the left OFD from the output of the right OFD (see equation

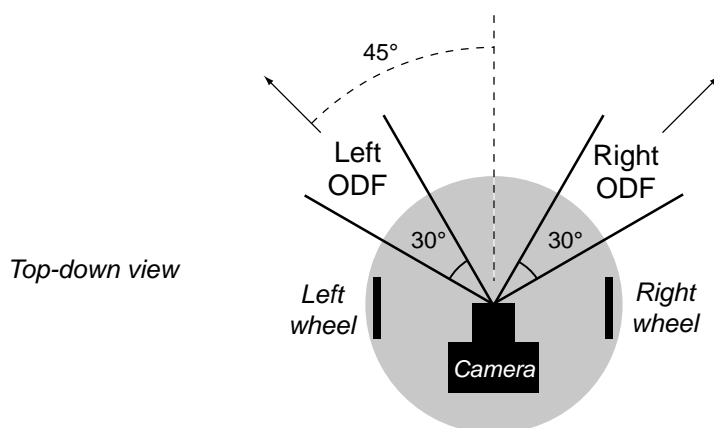


Figure 5.12: Arrangement of the OFDs on the Khepera equipped with frontal camera (see also figure 3.9a) for the obstacle avoidance experiment.

5.5).

As proposed in subsection 5.2.1, the steering control is composed of two states: (i) straight forward motion at constant speed (10cm/s) during which the system continuously computes  $OFDiv$ , (ii) rotation for a fixed amount of time (1s) during which sensory information is discarded. We chose one second in order to produce a rotation of approximately  $90^\circ$ , which is in accordance with what was observed in Tammero and Dickinson (2002b). The transition from state (i) to state (ii) is triggered whenever  $OFDiv$  reaches a threshold whose value is experimentally determined beforehand. The direction of the saccade is determined by the asymmetry  $OFDiff$  between left and right OFDs, i.e., the Khepera turns away from the side experiencing larger OF value.

Equipped with this steering control, the Khepera was able to navigate without collisions for more than 45 minutes (60'000 sensory-motor cycles), during which 84% of the time was engaged in straight motion and the remaining 16% in turning actions. Figure 5.13 shows a typical trajectory of the robot during this experiment and highlights the resemblance with flies' behaviour.

### 5.3.3 Steering Control of the Airplane

Encouraged by these results, we proceeded to autonomous steering experiments with the F2 airplane (subsection 3.1.3) in the arena depicted in figure 3.12e. The 30-gram aircraft is equipped with two miniature cameras oriented at  $45^\circ$  from the forward direction, each providing 28 pixels for the left and right OFDs spanning  $40^\circ$  (figure 5.14). How-

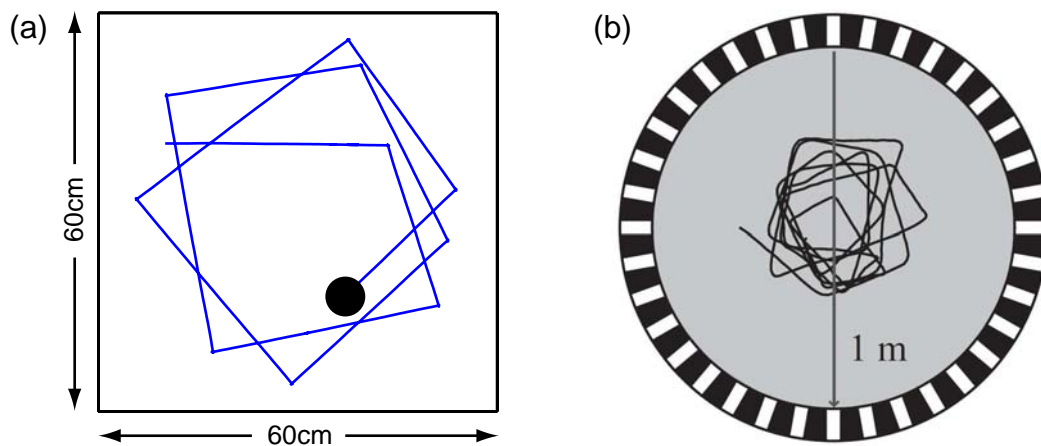


Figure 5.13: (a) Obstacle avoidance with the Khepera. Path of the robot in autonomous steering mode: straight motion with saccadic turning action when image expansion (*OFDiv*) reaches a threshold. The black circle represents the Khepera at its starting position. The path has been reconstructed from wheel encoders. Adapted from Zufferey and Floreano (2005) (b) For comparison, a sample trajectory (17s) within a textured background of a real fly *Drosophila melanogaster*. Reprinted from Tammero and Dickinson (2002b).

ever, at this stage we did not use yet vertical OFDs, but controlled the altitude of the airplane manually.

A radio connection (subsection 3.2.3) with a laptop computer is always established in order to log sensor data in real-time while the robot is operating. The plane is started manually from the ground by means of a joystick connected to the laptop. When it reaches an altitude of approximately 2m, a command is sent to the robot that puts it in autonomous steering mode. While in this mode, the human pilot has no access to the rudder (the vertical control surface, see figure 3.6), but can modify the pitch angle by means of the elevator (the horizontal control surface).<sup>6</sup> The typical sensory-motor cycle lasts 80ms. During this period, data from onboard sensors are processed, commands for the control surfaces are issued, and significant variables are sent to the laptop for later analysis. About 50% of this sensory-motor cycle is spent in wireless communication.

As a reminder, the control strategy is summarised in figure 5.15. During saccades, whose length has been set to 1 second<sup>7</sup>, the motor

<sup>6</sup>If required, the operator can switch back to manual mode at any moment, although a crash into the curtained walls of the arena does usually not damage the lightweight airplane.

<sup>7</sup>This length has been chosen in order to roughly produce 90° turns per saccade. However, this angle can fluctuate quite a bit depending on the velocity the robot

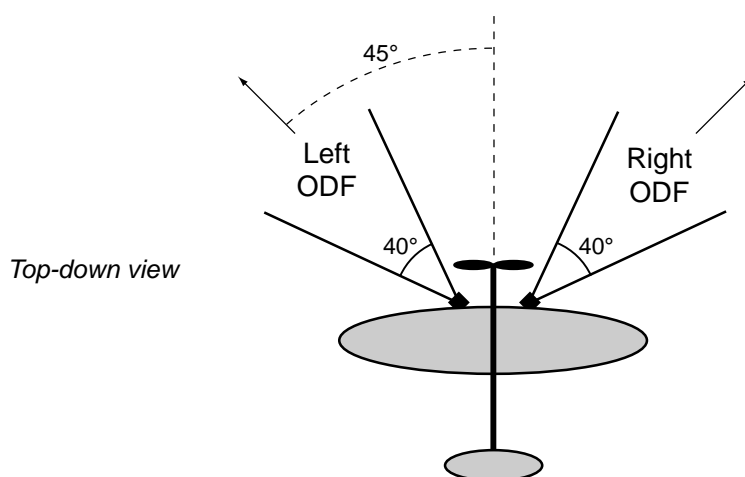


Figure 5.14: Arrangement of the two ODFs on the F2 airplane. See also the picture in figure 3.9b.

is set to full power, the rudder deflection follows an experimentally optimised curve up to full deflection, and the elevator is slightly pulled to compensate for the decrease in lift during banked turns (an action which is well known to air pilots). At the end of a saccade, the plane resumes straight flight while it is still in an inclined position. Since inclined wings always produce a yaw movement, the proportional controller based on the gyroscope (subsection 5.2.1) will compensate for the inclination and force the plane back to zero yaw and roll. We also implemented an “inhibition” period after the saccade, during which another turning action cannot be triggered. This allows for the plane to recover almost straight flight before deciding whether to perform another saccade. In our case, the inhibition is active as long as the gyroscope indicates an absolute yaw rotation larger than  $20^\circ/\text{s}$ . This inhibition period also allows to reset the *OFDiv* and *OFDiff* signals that can be affected by the strong optic flow values occurring just before and during the saccade.

Before testing the plane in autonomous conditions, the *OFDiv* threshold for initiating a saccade (see figure 5.9) has been experimentally determined by flying manually in the arena and recording OFD signals when frontally approaching a wall until the latest possible moment when the pilot had to start an emergency turn. The recorded OFD data were analysed and the threshold chosen on the basis of the value reached by *OFDiv* just before steering.

---

displays at the saccade start.

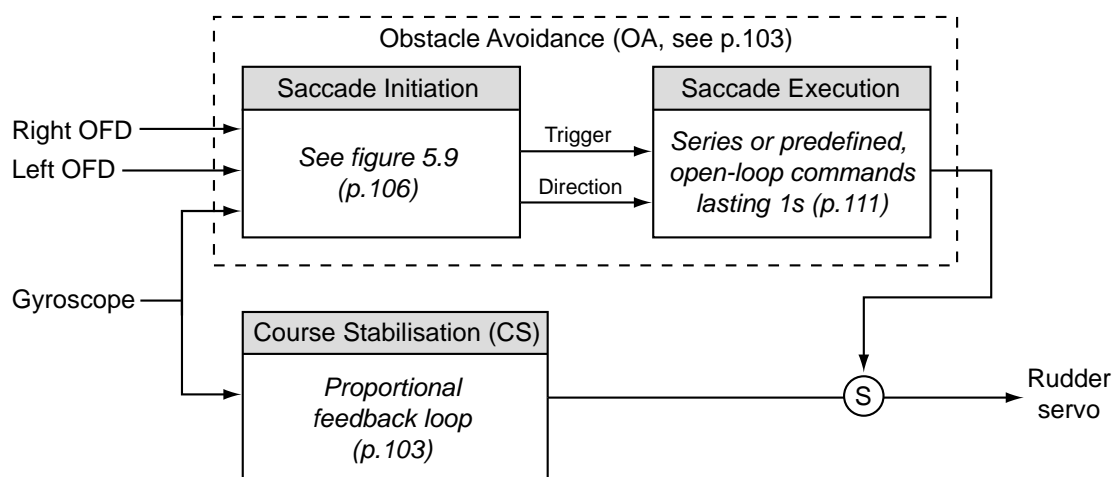


Figure 5.15: Overview diagram of the control strategy implemented on the F2 airplane. On the left are the sensory inputs (optic-flow detectors and gyroscope) and on the right is the system output (rudder actuator). This diagram is voluntarily inspired from the subsumption architecture proposed by Brooks (1999). The encircled S represents a suppressive node; this means that, when active, the signal coming from above replaces the signal usually going horizontally through the node.

## Results

In an endurance test, the 30-gram robot was able to fly collision-free in the 16x16m room for more than 4 minutes without any steering intervention.<sup>8</sup> The plane was engaged in turning actions only 20% of the time, which indicates that it flew always in straight trajectories except when very close to the walls. During those 4 minutes, it generated 50 saccades, and covered about 300m in straight motion.

Unlike the Khepera, the F2 has no embedded sensors allowing for plotting its trajectory. Instead, figure 5.16 displays a detailed 20-second sample of the data acquired during typical autonomous flight. Saccade periods are indicated with horizontal gray bars spanning all the graphs. In the first column, the gyroscope data provide a good indication of the behaviour of the plane, i.e., straight trajectories interspersed with turning actions, during which the plane can reach a yaw rotation rate of  $100^\circ/\text{s}$ . OF is estimated by the OFDs computed from the 1D images shown in the center column. The quality of the lightweight imagers and optics does not provide perfect images. As a result, OFD signals (on the two next graphs) are not always accurate,

<sup>8</sup>Video clips showing the behaviour of the plane can be downloaded from <http://phd.zuff.info>.

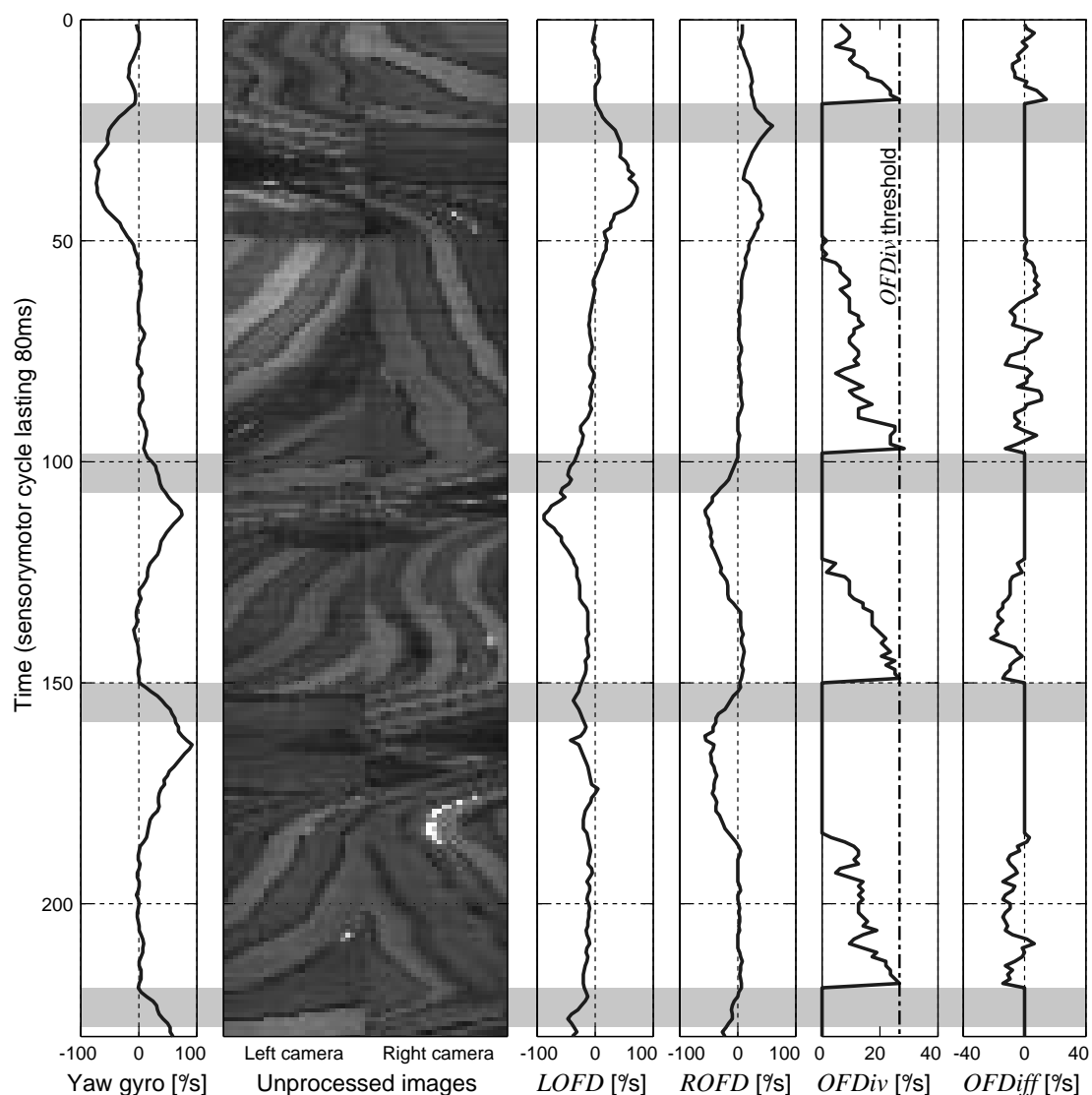


Figure 5.16: Sensor and OF data during autonomous flight (225 sensory-motor cycles). First column is the yaw gyroscope and indicates how much the plane is rotating (rightward positive). Next column displays the images as seen by the two cameras over time. Third and fourth columns are the OF as estimated by the left and the right OFDs, respectively. Fifth and sixth columns show OF divergence  $OFDiv$  and difference  $OFDiff$  when absolute value of the gyroscope is less than  $20^\circ/s$ , i.e., when the plane is flying almost straight. The dashed vertical line in the  $OFDiv$  graph represents the threshold for triggering a saccade. The gray horizontal lines spanning all the graphs indicate the saccades themselves, i.e., when the turning motor program is in action. The first saccade is leftward and the next three are rightward, as indicated by the gyroscope values in the first column. Adapted from Zufferey and Floreano (To appear).

especially when the plane is close to walls (few visible stripes) with a high yaw rotational velocity. This situation happens during the saccade inhibition period. Therefore, we set  $OFDiv$  and  $OFDiff$  (two last columns of Figure 5.16) to zero whenever the gyroscope indicates a yaw rotation larger than  $20^\circ/s$ .

When  $OFDiv$  reaches the threshold indicated by the dashed line, a saccade is triggered. The direction of the saccade is given by  $OFDiff$ , which is plotted on the right-most graph. The first turning action is leftward because  $OFDiff$  is positive when the saccade is triggered. The remaining turns are rightward because of the negative values of  $OFDiff$ . When the approach angle is not perpendicular, the sign of  $OFDiff$  is well defined, as in the case of the third saccade. In other cases, as before the second saccade,  $OFDiff$  is oscillating probably because the approach angle is almost zero. Note however that in such cases, the direction of the turning action is not important since the situation is symmetrical and there is no preferred direction for avoiding the obstacle.

## 5.4 Summary and Discussion

Bio-inspired, vision-based control strategies for autonomous steering and altitude control have been developed and assessed on a wheeled robot and an ultra-light aircraft. Information processing and navigation control are performed exclusively by the small embedded 8-bit microcontroller. In this section, we briefly discuss the steering and altitude control strategies and come back on the three levels of bio-inspiration.

### Steering Control

In comparison to most previous works in bio-inspired vision-based obstacle avoidance (subsection 1.2.2) our approach relies on less powerful processor and lower resolution visual sensor in order to enable operation in self-contained, ultra-light robots in real-time. In contrast to the optic-flow-based airplanes from Barrows et al. (2001) and Green et al. (2004) (see also subsection 1.2.2, second part), we demonstrated continuous steering over an extended period of time (50 avoidance actions without failure) with an aircraft that is able to avoid obstacles on both sides as well as frontal ones. However, we did not yet test our aircraft in other environments than the textured arena (figure 3.12e).

A question that could be raised about steering control experiments is why did the aircraft flown “only” 4 minutes<sup>9</sup> whereas the Khepera was able to ride for more than 45 minutes. The most probable reason of failure is the lighting conditions of the environment that was far less homogeneous than in the case of the Khepera (as can be seen in figure 3.12). In particular, it could happen that a camera was suddenly blind by light coming from outdoor. Since no brightness adaptation was implemented, this could provoke a failure of the saccade initiation mechanism. However, the 4 minutes of autonomous operation already demonstrate the good robustness that was achieved without much tuning of parameters. Due to time restriction (availability of the hall), it was indeed not possible to fine-tune all parameters (such as the time constants of the temporal low-pass filters) and to systematically analyse data just before a failure to determine exactly what happened.

### **Altitude Control**

Our approach to optic-flow-based altitude control proposes two new ideas with respect to previous work (Barrows et al., 2001; Chahl et al., 2004; Ruffier and Franceschini, 2004). The first one is the pitching rotatory optic-flow cancellation with a MEMS gyroscope, which allows to get rid of the spurious signals occurring whenever a pitch correction occurs. The second is the automatic tracking of the ground perpendicular distance (with a winner-take-all circuit choosing among several optic-flow detectors) releasing for the need of measuring the pitch angle with another sensor. Although much work remains to be done to further assess this idea<sup>10</sup>, the theoretical improvement is encouraging because there are not yet good solutions to provide a vertical reference on an ultra-light aircraft (MEMS inertial inclinometers are subject to longitudinal accelerations).

To cope with the issues of rotatory optic flow and vertical reference, Ruffier and Franceschini (2004) could regulate the absolute pitch angle

---

<sup>9</sup>Note that the airplane has a maximum energetic autonomy of only 30 minutes and the battery was not always fully charged at the moment of testing. Also, a human-operator must follow the aircraft during the whole testing period in order to send small altitude corrections and 45 minutes would require much endurance from the pilot.

<sup>10</sup>Testing different arrangement of optic-flow detectors with or without overlapping field-of-views, or explicitly using the information about the pitch within the control loop. Note that this method would likely require a quite high resolution of the optic-flow field and thus an high spatial frequency on the ground and a number of optic-flow detectors.



by means of the servomotor mounted at the articulation between the boom and the aircraft (see also figure 1.4c) and actively oriented the camera toward the ground (with a mechanism similar to gaze stabilization). Barrows et al. (2001) and Chahl et al. (2004) did not take care of this problem, which probably had a crucial influence on the limited results they obtained in altitude control (subsection 1.2.2).

Although the altitude control system has not yet been tested in flight, the strategy would be to regulate altitude only during straight motion in order to ensure an almost null roll angle. During saccade, open-loop control of the elevator command has proved to be acceptable to keep altitude since the duration of the saccade is very limited. Optic-flow-based altitude control during banked turns would anyway not be viable because of considerable changes in flight velocity and large roll angles that would preclude a vertically oriented (with respect to the body-fixed frame) optic-flow sensor from measuring the perpendicular distance to the ground.

### **Bio-inspiration at Three Levels**

Navigation control of our robots was enabled by selectively mimicking flying insects at different levels (chapter 2): perceptive organs, information processing and behaviour. Here we come back on these three topics and discuss the similarities and differences between our robots and the inspiring animals.

The perceptive organs of flying insects has been our main source of inspiration in the selection of sensors for the robots. Although the flies possess a range of mechanosensors (subsection 2.2.3), eyes and halteres are the most important sensors for flight control (section 2.2). It is remarkable that, unlike most classical autonomous robots, flying insects possess no active distance sensors like sonars. This is probably because of the inherent complexity and energy consumption of such sensors. The gyroscope equipping the robot can be seen as a close copy of the Diptera's halteres (subsection 2.2.2). The selected artificial vision system (subsection 3.2.2) shares with its biological counterpart an amazingly low resolution. Its inter-pixel angle ( $1.4\text{-}2.6^\circ$ ) is of the same order of magnitude as the interommatidial angle of most flying insects ( $1\text{-}5^\circ$ , see subsection 2.2.1). On the other hand, the field of view of our robots is much smaller than that of most flying insects. This discrepancy is mainly due to the lack of technology allowing for building miniature, omnidirectional visual sensors sufficiently light to

fit the constraints of our indoor aircrafts. In particular, no industrial interest exists so far in the development of a compound lens architecture and commercial omnidirectional mirrors are too heavy. We have partly compensated the lack of omnidirectional vision sensors by using two or more small vision sensors looking in the directions of interest, which were carefully identified beforehand (section 5.1). In principle, three 1D cameras (two horizontal, pointing forward at  $45^\circ$ , and one longitudinally oriented, pointing vertically to the ground) should be sufficient for autonomous steering and altitude control of an airplane in simple indoor environments.

The second level of bio-inspiration concerns the stage of information processing. Although the extraction of OF itself is not inspired by the EMD model (subsection 2.3.2) because of its known dependency on contrast and spatial frequency (subsection 4.2.1), OF is employed in our robots as a primary cue for behaviour control. An efficient algorithm for OF detection has been adapted to fit the embedded 8-bit microcontroller (section 4.2). As in flying insects, divergent optic flow is used to detect approaching objects (subsection 2.3.3) and ventral optic flow as a cue to perceive altitude over ground (subsection 2.4.4). The attractive feature of such simple solutions for depth perception is that they do not require explicit measurement of distance or time-to-contact, nor do they rely on accurate knowledge of the flight velocity. Furthermore, we have shown that in certain cases, they intrinsically adapt to the flight situation by triggering warnings farther away from obstacles that appear to be harder to avoid (subsection 5.1.1). Another example of bio-inspired information processing is the fusion of gyroscopic information with vision. Although the simple scalar summation employed in our robots is probably far from what actually happens in the fly's nervous system, it is clear that some important interactions between visual input and halteres' feedback exist in the animal (subsection 2.3.3).

The third level of bio-inspiration concerns the behaviours. Altitude control is based upon mechanisms inferred from experiments with honeybees that have been shown to regulate the experienced OF in a number of situations (subsection 2.4.4). Instead, the steering strategy composed of straight sequences interspersed with rapid turning actions is inspired by flies' behaviour. While in flies some saccades are spontaneously generated in the absence of any visual input, reconstruction of OF patterns based on flies' motion through an artificial visual landscape suggests that image expansion plays an fundamental

role in triggering saccades (subsection 2.4.3). Apart from providing a way of minimising rotatory optic flow, straight flight sequences also increase the quality of visual input by maintaining the plane horizontal and spare flight energy (subsection 5.2.1). In our robots, the entire saccade is performed without sensory feedback. During saccades, biological EMDs are known to operate beyond their linear range where the signal could even be reversed because of temporal aliasing (Srinivasan et al., 1999)). However, the role of visual feedback in the control of these fast turning manoeuvres is still under investigation (Tammero and Dickinson, 2002a). Halteres' feedback is more likely to have a major impact on the saccade duration (Dickinson, 1999). Our robots do not yet use any sensory feedback during saccade. However, the use of gyroscopic information could provide a good way of controlling the angle of the rotation. Finally, the precise roles of halteres and vision in course (or gaze) stabilisation of flies is still unclear (subsection 2.4.2). Both sensory modalities are believed to have an influence, whereas in our robots, course stabilisation and RotOF cancellation (which can be seen as the placeholder of gaze stabilisation in flies) rely exclusively on gyroscopic information.

Finally, bio-inspiration was of great help in the design of autonomous, vision-based robots. However, a lot of engineering insight was required to tweak biological principles so that they could match the final goal. It is also to notice that biology often lack of synthetic models, sometimes because of the lack of engineering attitude in biologists (see Wehner, 1987 for a discussion), sometimes because of the lack of experimental data. For instance, biologists are just starting to study neuronal computation in flies with natural, behaviourally relevant stimuli (Lindemann et al., 2003), which will probably question the principles established so far with simplified stimulus (Egelhaaf and Kern, 2002). Moreover, mechanical structures of flying robots as well as their processing hardware will never perfectly match biological systems. These considerations compelled us to test an alternative solution to biomimetism as performed in this chapter, which takes inspiration from biology at (yet) another level (next chapter).



# Chapter 6

## Evolutionary Approach

In this chapter things get slightly out of hand. You may regret this, but you will soon notice that it is a good idea to give chance a chance in the further creation of new brands of vehicles. This will make available a source of intelligence that is much more powerful than any engineering mind.

---

V. Braitenberg (1984)

### Abstract

This chapter explores alternative strategies for vision-based navigation that meet the constraints of our indoor flying robots (few computational resources, only low-resolution vision and gyroscope, and complex dynamics). A genetic algorithm is used to evolve artificial neural networks that map sensory signals into motor commands. A simple neural network model has been developed, which fits the limited processing power of our lightweight robots and ensures real-time capability. The same sensory modalities as in chapter 5 are used, whereas information processing strategies and behaviours are automatically developed through artificial evolution. First tested on wheels with the Khepera, this approach results in successful vision-based navigation, which does not rely on optic flow. Instead, the evolved controllers simply measure the image contrast rate to steer the robot. Building upon this result, neuromorphic controllers are then evolved for steering the Blimp2b, resulting in efficient trajectories maximising forward translation while avoiding wall contacts and coping with unusual situations.

## 6.1 Method

### 6.1.1 Rationale

One of the major problems faced by engineers willing to use bio-inspiration in the process of hand-crafting artificial systems is the overwhelming amount of details and varieties of biological models. In the previous chapter, we selected and adapted the principles of flying insects that seemed most relevant to our goal of designing autonomous robots. However, it is not obvious that the use of optic flow as visual preprocessing is the only alternative for these robots to navigate successfully. Moreover, the navigation strategy using sequences of straight movements and saccades is equally questionable. It may be that other strategies are well adapted to the sensors, processing resources, and dynamics of the robots.

This chapter is an attempt to let open the question of how visual information should be processed, and what is the best strategy to fulfil the initial requirement of "maximising forward translation", without dividing it into predefined behaviours such as course stabilisation, obstacle avoidance, etc. (subsection 1.1.1). To achieve that, we use the method of evolutionary robotics (ER, see description in subsection 1.2.3). This method allows to define a generic controller (*neural network*<sup>1</sup>) containing free parameters (*synaptic weights*) that are adapted to satisfy a performance criterion (*fitness function*) while the robot is moving in its environment. In our application, the interest of this method is threefold:

- It allows to fit the embedded microcontroller limitations (no floating point, limited computational power) by designing adapted *artificial neurons* (computational units of a neural network) before using evolution to interconnect them.
- It allows to specify the task of the robot ("maximising forward translation") by means of the fitness function while avoiding to specify the details of the strategies that should be used to accomplish this task.

---

<sup>1</sup>Although other types of control structures can be used, the majority of experiments in ER employ some kind of artificial neural networks because they offer a relatively smooth search space and are a biologically plausible metaphor of mechanisms that support animal behaviours (Nolfi and Floreano, 2000).

- It takes implicitly into account the sensory constraints and dynamics of the robots by measuring their fitness while they are actually moving in the environment.

The drawback of ER with respect to hand-crafting bio-inspired controllers is that it requires a large amount of evaluations of randomly initialised controllers. To cope with this issue, we first rely on the Khepera robot (subsection 3.1.1), which is able to support any type of random control, withstand shocks against walls, and is externally powered (does not rely on battery). This wheeled platform allows us to test and compare different kinds of visual preprocessing and parameters of evolution. The next step consists in building upon the results obtain on wheels to tackle the more complex dynamics of flying robots. Since the F2 airplane (subsection 3.1.1) cannot support random control, nor has yet an accurate simulator (subsection 3.3.2), we use the Blimp2b (subsection 3.1.2) as an intermediate flying platform that already features much more complex dynamics than the Khepera robot. Moreover, a complete airship dynamic model has been developed (appendix A) that enables accurate simulation and faster evolutionary experiments.<sup>2</sup> Since obtaining good solutions in simulation is not a goal *per se*, evolved controllers are systematically tested on the real Blimp2b at the end of the evolutionary process.

In addition to maximising forward translation, these two platforms (Khepera and Blimp2b) enables the study of a corollary aspect of basic navigation: “how getting out of critical situations such as facing a wall or a corner”. This could not be tackled in the previous chapter because (i) the F2 could not be positioned in such a situation without resulting in an immediate crash and (ii) optic flow provides no information when the robot is not in motion. The robots selected as testbeds in this chapter are both able to stop and reverse their course. An interesting question is thus whether evolved controllers can manage that kind of critical situations and, if so, what visual cues do they use. Note that in order to tackle this issue, there is no need for modifying the global performance criterion of “maximising forward translation”. It should be sufficient to start each evaluation period with the robot in such a critical situation. If the robot cannot quickly get out, it will not be able to move forward during the rest of the evaluation period.

---

<sup>2</sup>40 to 50 times faster than real-time (subsection 3.3.2).

### 6.1.2 Evolutionary Process

An initial *population* of different *individuals*, each represented by its *genetic string* encoding the parameters of a neural controller, is randomly created. The individuals are evaluated one after the other on the same physical (or simulated) robot. In our experiments, the population is composed of 60 individuals. After ranking the individuals according to their performance (using the fitness function, see subsection 6.1.4), each of the top 15 individuals produces 4 copies of its genetic string in order to create a new population of the same size. The individuals are then randomly paired for *crossover*. One-point crossover is applied to each pair with probability 10% and each individual is then mutated by switching the value of a bit with probability of 1% per bit. Finally, a randomly selected individual is substituted by the original copy of the best individual of the previous generation (*elitism*). This procedure is referred to as a rank-based truncated selection, with one-point crossover, bit *mutation*, and elitism (Nolfi and Floreano, 2000).

Each individual of the population is evaluated on the robot for a certain amount  $T$  of sensory-motor cycles (each lasting 50 to 100ms). The length of the *evaluation period* is adapted to the size of the arena and the typical robot velocity, so that individuals have a chance to experience a reasonable amount of situations. In this thesis, the length of the evaluation period is usually within 40-120 seconds (or 400 to 2400 sensory-motor cycles). Usually, at least two evaluations are carried out with the same individual in order to average the effect of different starting positions on the global fitness.

This evolutionary process is handled by the software *goevo* (subsection 3.3.1), which manages the population of genetic strings, decodes each of them into an individual with its corresponding neural controller, evaluates the fitness and does the selective reproduction at the end of the evaluation of the whole population. Two operational modes are possible (see appendix B.2 for more details). In the *remote mode*, the neural controller is emulated within *goevo*, which exchanges data with the robot at every sensory-motor cycle. In the *embedded mode*, the neural controller is implemented within the microcontroller of the robot and data exchange occurs only at the beginning and at the end of the evaluation period. The remote mode allows to monitor the internal state of the controller whereas the embedded mode ensures full autonomy of the robot at the end of the evolutionary process.



### 6.1.3 Neural Controller

An artificial *neural network* is a collection of units (*artificial neurons*) linked by weighted connections (*synapses*). Input units receive sensory signals and output units control actuators. Neurons that are not directly connected to sensors or actuators are called *internal units*. In its most simple form, the output of an artificial neuron  $y_i$  (also called *activation value* of the neuron) is a function  $\Lambda$  of the sum of all incoming signals  $x_j$  weighted by *synaptic weights*  $w_{ij}$ :

$$y_i = \Lambda\left(\sum_j^N w_{ij}x_j\right), \quad (6.1)$$

where  $\Lambda$  is called the *activation function*. A convenient activation function is  $\tanh(x)$  because whatever the sum of the input is, the output remains in the range  $[-1,+1]$ . This function acts as a linear estimator in its center region (around zero) and as a threshold function in the periphery. By adding an incoming connection from a *bias unit* with a constant activation value of  $-1$ , it is possible to shift the linear zone of the activation function by modifying the synaptic weight from this bias.

Section B.1 in appendix describes how the artificial neural network is implemented in the 8-bit microcontroller. This embedded implementation is called PIC-NN (PIC compatible neural network). The PIC-NN is a discrete-time, recurrent neural network, whose computation is executed once per sensory-motor cycle. Recurrent and lateral connections take the pre-synaptic activation values from the previous cycle as input. The number of input and internal units, the number of direct connections from input to output, and the activation of lateral and recurrent connections can be freely chosen. In all experiments presented in this chapter, the PIC-NN has 2 internal neurons and 2 output neurons, whose activation values are directly used to control the actuators of the robot (positive values correspond to positive rotation of the motor, whereas negative values yield negative rotation). The two internal neurons are inserted in the hope they could serve as a stage of analysis or summary of the incoming visual input in order to provide the output layer with more synthetic signals. Recurrent and lateral connections are enabled only in the output layer and enable an inertia or low-pass filtering effect on the signals driving the motors. The number of input units depends on the type of sensory preprocessing, which will be described for each experiment.

Each synapse of a PIC-NN is encoded on 4 bits (section B.1). The

corresponding binary genetic string is thus simply composed of the juxtaposition of the 4-bit blocks.

### 6.1.4 Fitness Function

The design of a *fitness function* for the evaluation of the individuals is a central issue to any evolutionary experiment. In this chapter, we employ a fitness function that is measurable using sensors available on the robots and sufficiently simple to avoid unwanted pressure toward specific behaviours (e.g., sequences of straight movements and rapid turning actions near walls). The fitness is simply a measure of forward translation.

With the Khepera, the instantaneous fitness is the average of the wheel speeds (based on wheel encoders):

$$\Phi_{Khepera}(t) = \begin{cases} (v_L(t) + v_R(t))/2 & \text{if } (v_L(t) + v_R(t)) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (6.2)$$

where  $v_L$  and  $v_R$  are the left and right wheel speeds, which are normalised with respect to their maximum allowed rotation rate (corresponding to a forward motion of 12cm/s). If the Khepera rotates on the spot (i.e.,  $v_L = -v_R$ ), the fitness is null. If only one wheel is set to full forward velocity, while the other one remains blocked, the fitness reaches 0.5. When the Khepera tries to push against a wall, its wheels are blocked by friction, resulting in null fitness.

In order to measure forward translation of the Blimp2b, we use the anemometer located below its gondola (figure 3.2). The instantaneous fitness is thus expressed as:

$$\Phi_{Blimp}(t) = \begin{cases} v_A(t) & \text{if } v_A(t) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (6.3)$$

where  $v_A$  is the output of the anemometer, which is proportional to the forward speed (the direction where the camera is pointing).  $v_A$  is also normalised with respect to the maximum value obtained during straight motion at full speed. Particular care has been taken to ensure that the anemometer is outside the flux of the thrusters so that it does not rotate when the blimp is pushing against against a wall. Also, no significant rotation of the anemometer is observed when the blimp rotates on the spot.

The instantaneous fitness values given in equations (6.2) and (6.3)

are then averaged over the entire evaluation period:

$$\bar{\Phi} = \frac{1}{T} \sum_{t=1}^T \Phi(t). \quad (6.4)$$

where  $T$  is the number of sensory-motor cycles of a trial period. A fitness of 1.0 thus corresponds to a robot moving straight forward at maximum speed for the entire duration of the evaluation period. However, this cannot be achieved in our experimental environment (figure 3.12) where the robots must often turn to avoid walls.

## 6.2 Experiments on Wheels

In this section, we first apply the method with the Khepera to determine whether evolution can produce efficient behaviour when the neural controller is fed with raw vision and compare the results to the case when optic-flow is provided instead. Then, we tackle the problem of coping with critical situations. The results will serve as a basis for evolutionary experiments with the Blimp2b (section 6.3).

All the experiments in this section are carried out on the Khepera equipped with the kevopic extension turret and the frontal 1D camera in the 60x60cm textured arena (chapter 3). An evaluation period lasts 40 seconds (800 sensory-motor cycles of 50ms) and is repeated two times per individual. The fitness of the two evaluation periods are then averaged. The resulting fitness graphs are based on an average across 3 evolutionary runs starting from a different random initialisation of the genetic strings.

### 6.2.1 Raw Vision versus Optic Flow

To answer the questions of whether optic flow and/or saccadic behaviour are required (subsection 6.1.1), two comparative experiments are set up. In the first one, called “raw vision”, the entire image is fed to the neural controller without any temporal filtering<sup>3</sup>, whereas in the second, called “optic flow”, four optic flow detectors (OFDs, see subsection 4.2.5) serve as exclusive visual input to the neural controller (figure 6.1). The initialisation procedure before each evaluation period consists in a routine where the Khepera drives away from the walls for

<sup>3</sup>As opposed to optic-flow processing, which is a spatio-temporal operation (see equations 4.5 and 4.8).

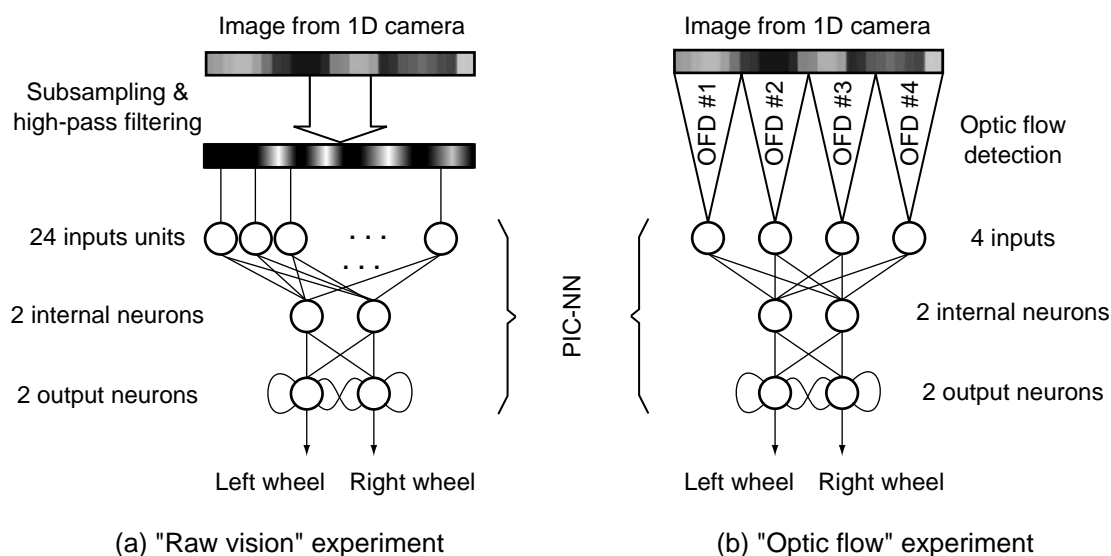


Figure 6.1: Configuration of visual preprocessing and PIC-NN for the comparison between “raw vision” and “optic flow”. (a) The 50 pixels from the 1D camera are subsampled to 25 and high-pass filtered with a rectified spatial difference among every two neighbouring pixels. The resulting 24 values are directly send to the 24 inputs of the PIC-NN. (b) The 48 pixels are divided into 4 regions of 12 pixels, on which the image interpolation algorithm (I2A, see subsection 4.2.3) is applied. The optic flow detectors (OFDs) outputs are then passed on to the 4 inputs of the underlying PIC-NN.

5 seconds using its proximity sensors (subsection 3.1.1). This avoids to deal with the corollary question of whether evolved individuals can manage critical situations, which will be tackled in the next subsection.

The first experiment with “raw vision” capitalises on existing results and is directly inspired by the experiment reported by Floreano and Mattiussi (2001), where a *Khepera* was evolved for vision-based navigation in the same kind of textured arena. The main difference between this experiment and the one presented in this thesis concerns the type of neural network.<sup>4</sup> The controller used by Floreano and Mattiussi (2001) was a spiking neural network emulated in an off-board computer (remote mode, see subsection B.2)<sup>5</sup> instead of a PIC-NN. The idea of high-pass filtering vision before passing it on to the neural network has been maintained in this experiment, although the processing

<sup>4</sup>Other minor differences concern the vision module (see Zufferey et al., 2003), the number of pixels used (16 instead of 24), the details of the fitness function, and the size of the arena.

<sup>5</sup>More recent experiments demonstrated the use of simpler spiking networks for embedded computation in a non-visual task (Floreano et al., 2002). See Floreano et al. (2003) for a review.

is done slightly differently in order to reduce computational cost.<sup>6</sup> The main reason for high-pass filtering visual input was to reduce dependency on background light intensity.<sup>7</sup>

In the second experiment with optic-flow, the parameters remain unchanged, except the visual preprocessing and the number of input units in the PIC-NN. Note that the two external OFDs have exactly the same configuration as in the optic-flow based steering experiment (see figure 5.12). Therefore, this visual information together with a saccadic behaviour should, in principle, be enough to steer the robot efficiently in the experimental arena.

## Results

The graph in figure 6.2a shows the population mean and best fitness over 30 generations for the case of “raw vision”. The fitness rapidly improves in the first 5 generations and then gradually reaches a plateau of about 0.8 around the 15th generation. This means that evolved controllers found a way of moving forward and avoiding to get stuck against surrounding walls. However, the fitness value does not tell us about the specific behaviour adopted by the robot. To answer this question, the best evolved controller was tested and its wheel encoders recorded in order to reconstruct the trajectory (figure 6.2b). The robot moves along a looping trajectory, whose curvature depends on the visual input.<sup>8</sup> Note that this behaviour is not symmetrical. Evolution finds a strategy consisting in turning always in the same direction (which can vary from experiments to experiments) and adapting the curvature radius to exploit the available arena space. In this experiment, the best evolved controllers always set their right wheel to full speed, and control only the left one to steer the robot. This strategy is in contrast with the hand-crafted solution implemented in chapter 5, which consisted in going straight and avoiding walls at the last moment and in the direction opposed to the closest side.

With "optic flow" as visual input, the resulting fitness graph (fig-

---

<sup>6</sup>Instead of implementing a Laplacian filter with a kernel of 3 pixels  $[-.5 \ 1 \ -.5]$ , here we use a rectified spatial difference of each pair of neighbouring pixels, i.e.,  $|I(n) - I(n - 1)|$ , where  $n$  is the pixel index and  $I$  the intensity. The outcome is essentially the same, since both filters provide a measure of local image gradient.

<sup>7</sup>Although experimental arenas are artificially lit, they are not totally protected from natural outdoor light. Background light intensity can thus fluctuate between day and night.

<sup>8</sup>Resulting behaviour is very similar to that obtained by Floreano and Mattiussi (2001).

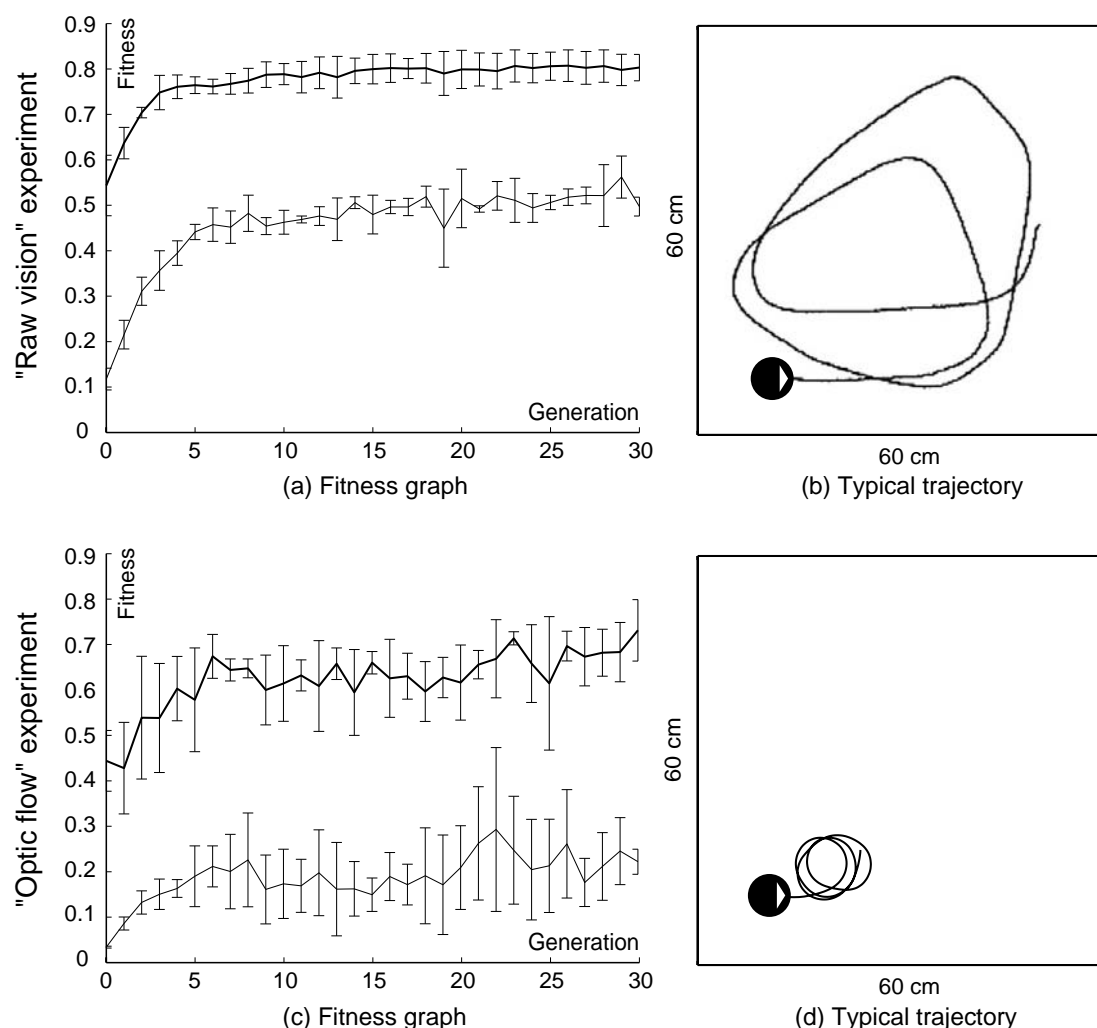


Figure 6.2: Comparative results of the “raw vision” and “optic flow” experiments. (a & c) Population mean (thin line) and best (thick line) fitness for 30 generations. Data points are averages over three evolutionary runs and error bars are the standard deviations among those three runs. (b & d) Typical trajectory of the best individual is plotted based on data from the wheel encoders.

Figure 6.2c) displays lower maximum values than in the previous experiments. The resulting trajectory (figure 6.2d) reveals that only very minimalist solution is found, where the robot rotates in small circles. This is even not vision-based navigation, because visual input does not have any influences on the constant turning radius. This strategy can still produce a relatively high fitness of almost 0.7 because individuals were always initialised far from the walls at the beginning of the evaluation periods and had thus some space to move like this, independently of their initial heading.

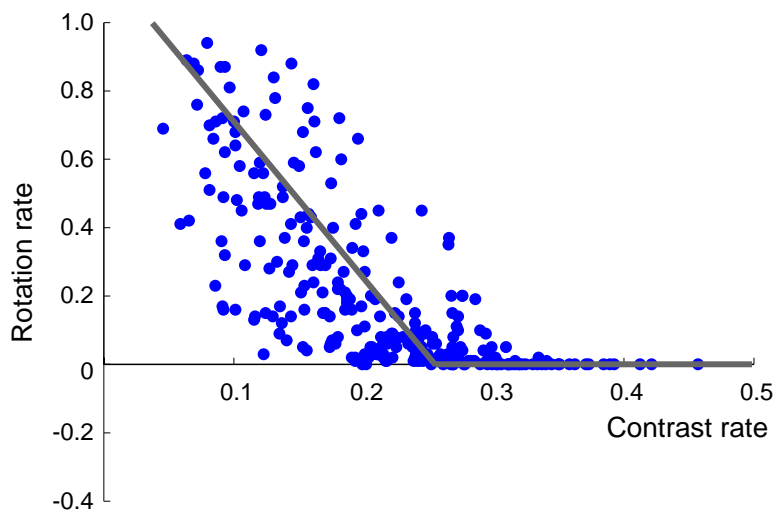


Figure 6.3: Khepera rotation rate versus image contrast rate during normal operation of the best evolved individual in its experimental arena. The contrast rate is the spatial average of the high-pass filter output (a value of 1.0 would correspond to an image composed exclusively of alternately black and white pixels). The rotation rate is given by  $(v_L - v_R)$ , where  $v_L$  and  $v_R$  are normalised in the range  $[-1, +1]$ .

### Analysis

Evolution with optic-flow as visual preprocessing does not produce acceptable navigation strategies, although the neural controller is provided with the same kind of visual input as in subsection 5.3.2. This can be explained by the fact that OFDs give useful information only when the robot is moving in a particular manner (straight forward at almost constant speed), but since the output of the neural networks used here depend solely on the visual input, it is likely that a different neural architecture is needed to properly exploit information from optical flow.

In contrast, evolution with “raw vision” produces interesting results with this simple PIC-NN. In order to understand how the visual information could be used by the neural network to produce the efficient behaviour, we made the hypothesis that the controller relies essentially on the contrast rate present in the image (a spatial sum of the high-pass filtered image). To test this hypothesis, we plotted the rotation rate  $(v_L - v_R)$  as a function of the spatial average of the visual input (after high-pass filtering) over the entire field of view (FOV) while the individual was freely moving in the arena. The resulting graph (figure 6.3) shows that an almost linear relation exists between the contrast

rate over the entire image and the rotation rate of the Khepera. In other words, the robot tends to move straight when a lot of contrast is present in the image, whereas it increases its turning rate as soon as less contrast is detected. The dispersion of the points in the right part of the graph shows that the processing of this particular neural network can not be fully explained by this strategy. In particular, it is likely that some parts of the image are given more importance than other ones in the steering process. However, this simple analysis reveals the logic of the evolved strategy, which can be summarised as follows: “move straight when contrast rate is high, and increase turning rate linearly with decreasing contrast rate” (see the thick gray lines in figure 6.3).

In summary, rather than relying on optic flow and symmetrical saccadic obstacle avoidance, the successful controllers employ a purely spatial property of the image (the contrast rate) and produce smooth trajectories to circumnavigate the arena only in one direction.

### 6.2.2 Coping with Critical Situations

This subsection tackles the issue of the critical situations occurring when the robot is facing a wall (or a corner) by adopting a set of additional precautions during the evolutionary process. At the same time, we build upon the previous results in order to decrease the number of sensory input to the PIC-NN. This should allow to decrease the size of the genetic string and accelerate the evolutionary process.

#### Additional Precautions

In order to force individuals to cope with critical situations without fundamentally changing the fitness function, a set of three additional precautions are taken:

- Instead of driving the robot away from walls, the initialisation procedure places them against a wall by driving them straight forward until one of the front proximity sensor becomes active.
- The evaluation period is prematurely interrupted (after 5 seconds) if the individual has not reached at least 10% of the maximum fitness (i.e., 0.1) at that time.
- The instantaneous fitness function  $\Phi(t)$  is set to zero whenever a proximity sensor (which have a limited range of about 1-2 cm) is active.



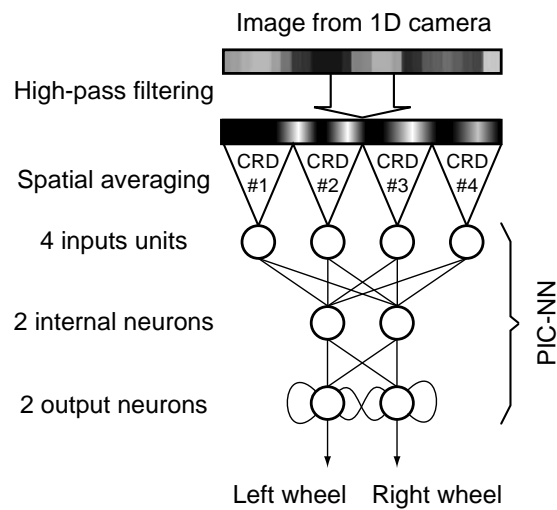


Figure 6.4: Visual preprocessing and PIC-NN for the experiment with critical starting situations. The intensity values from the 1D camera are first high-pass filtered with a rectified spatial difference among every two neighbouring pixels. The spatial averaging over 4 evenly distributed regions occurs in order to feed the 4 input units of the PIC-NN.

### Visual Preprocessing

Since individuals evolved with access to the entire image mainly relied on global contrast rate in the image (see analysis in subsection 6.2.1), here we deliberately divide the image into 4 regions evenly distributed and compute the contrast rate over each of them before feeding the neural controller with the resulting values (figure 6.4). We call *contrast rate detector* (CRD) this kind of preprocessing associated with its corresponding image region. Since the high-pass spatial filtering is a kind of edge enhancement, the output of such a CRD is essentially proportional to the number of edges seen in the image region. This preprocessing allows to reduce the size of the neural network with respect to the “raw vision” approach and thus limit the search space of the genetic algorithm<sup>9</sup>. Since the additional precautions make the task more complex, the reduction of the search space is not expected to yield significant acceleration in the evolutionary process. However, this will help to maintain the number of required generations to a reasonable amount.

<sup>9</sup>The genetic string encoding the PIC-NN measures 80 bits instead of 240 bits in the “raw vision” experiment.

## Results

The resulting fitness graph (figure 6.5a) is similar to the one of the “raw vision” experiment (figure 6.2a). Although progressing slightly slower in the first generations, the final maximum fitness values of 0.8 after 30 generations are identical. The increased difficulty of the task due to the additional precautions is indicated by the fitness graph by the lower average fitness over the population (approx. 0.35 instead of 0.5).

The genetic algorithm found a way of coping with the new set of precautions in spite of the limited number of sensory inputs. In order to better demonstrate the higher robustness obtained in this experiment, the typical trajectories of the best evolved individuals of different evolutionary runs are plotted with the Khepera starting against a wall (and facing it). We observe a number of different behaviours that produce the same average fitness values. In all cases, the individuals manage to quickly escape from the critical starting position, either by backing away from the wall (figure 6.5b-c) during a short period of time (roughly 2 seconds) or rotating on the spot until finding a clear path (figure 6.5d). Once escaped, they quickly recover a preferred trajectory yielding high fitness. The behaviours can either consist of navigating in large circles and slightly adapting the turning rate when necessary (figure 6.5b), or moving in straight segments and steering only when close to a wall. In this latter case, the individuals either describe smooth turns (figure 6.5c) or on-the-spot rotations (figure 6.5d). The individuals that rotate on the spot when facing a wall exploit the same strategy in order to avoid collisions later on.

Those results demonstrate that a range of possible strategies exist, which equally fulfil the basic requirement of “maximising forward translation” even if the starting position is critical (i.e., requires specific behaviour that is not always used afterwards). Rather than using optic-flow, those strategies rely on spatial properties (contrast rate) of the visual input.

## 6.3 Aerial Experiments

A preliminary set of experiments entirely carried out on a physical blimp (Zufferey et al., 2002) indicated that artificial evolution can generate, in about 20 generations, neuromorphic controllers able to drive the Blimp1 (ancestor of the current Blimp2b, less manoeuvrable due to another arrangement of the thrusters and not equipped with a gyro-

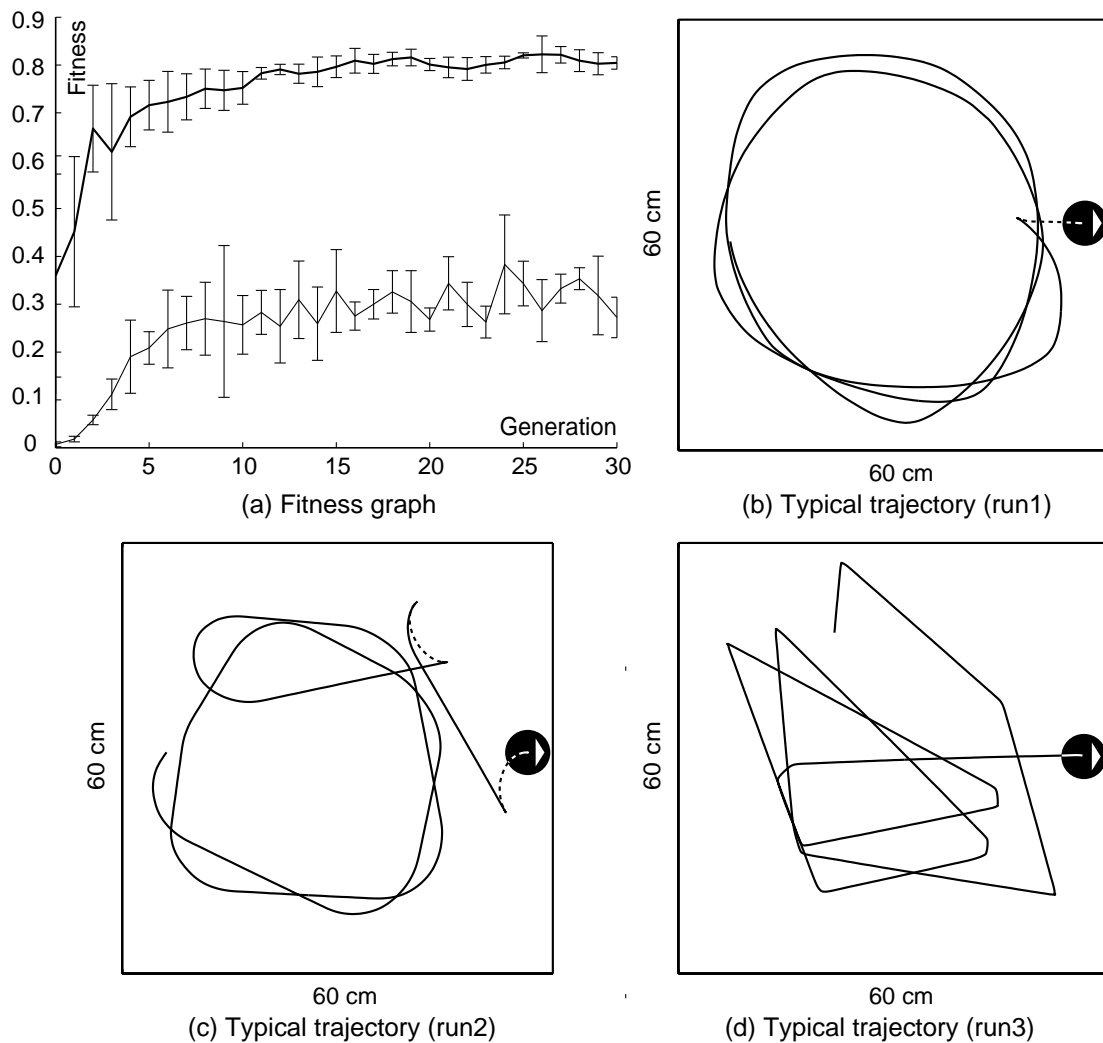


Figure 6.5: Results of the evolutionary experiment with the Khepera using 4 contrast rate detectors and coping with critical starting situations. (a) Population mean (thin line) and best (thick line) fitness for 30 generations. Data points are averages over three evolutionary runs. (b-d) Typical trajectories of the best individuals of the 3 runs. The Khepera (black circle with the white arrow indicating the forward direction) is always placed perpendicularly facing a wall at the beginning to demonstrate its ability to get rapidly out of this difficult situation. A dotted trajectory line indicates backward motion.

scope) around the textured arena. The obtained strategies largely relied on contacts with walls to stabilise the course of the blimp in order to gain forward speed. Later on, a redesigned platform (Blimp2, very similar to the Blimp2b) equipped with a yaw gyroscope (which output was passed on to the neural controller) produced smoother trajectories without using the walls for stabilisation (Floreato et al., 2005). Those evolutionary runs carried out directly on the physical flying robots were

rather time-consuming. Only 4 to 5 generations could be evaluated in 1 day (the battery had to be changed every 2-3 hours) and more than 1 week was required to obtain successful controllers. Additionally, some runs had to be cancelled because of mechanical problems such as motor deficiencies.

After these preliminary experiments, the simulator (subsection 3.3.2) has been developed in order to accelerate and facilitate the evolutionary runs. In contrast to previous experiments with Blimp1 and Blimp2, here we present experiments with the Blimp2b where

- evolution is entirely performed in simulation and only the best evolved controllers are transferred to the real robot,
- the same set of precautions as developed with the Khepera (subsection 6.2.2) are used to force individuals to cope with critical situations (facing a wall or a corner),
- a set of virtual<sup>10</sup> proximity sensors are used during simulated evolution to set the instantaneous fitness to zero whenever the blimp is close to the wall (part of the above-mentioned precautions).

This section is divided into two parts. First the results obtained in simulation are presented, then the transfer of the best evolved individual to reality is described.

### 6.3.1 Evolution in Simulation

The neural controller is evolved in order to steer the Blimp2b in the square arena (figure 3.12d) using only visual and gyroscopic information available from on-board sensors.<sup>11</sup> As for the latest experiment with the Khepera (subsection 6.2.1), visual input is preprocessed with 4 CRDs, which feed the PIC-NN (figure 6.6). In addition, the pixel intensities coming from the 1D camera are binarised. Since the visual surrounding both in simulation and reality is black and white, thresholding the image allows to ensure a better match among the two worlds.

Since one of the big differences between the Khepera and the Blimp2b is the need for course stabilisation (CS, see table 3.2), the

---

<sup>10</sup>We call “virtual” a sensor that is only implemented in simulation, but does not exist on the real robot.

<sup>11</sup>In these experiments, altitude is not under evolutionary control, but is automatically regulated using information from the distance sensor pointing downward (see subsection 3.1.2).

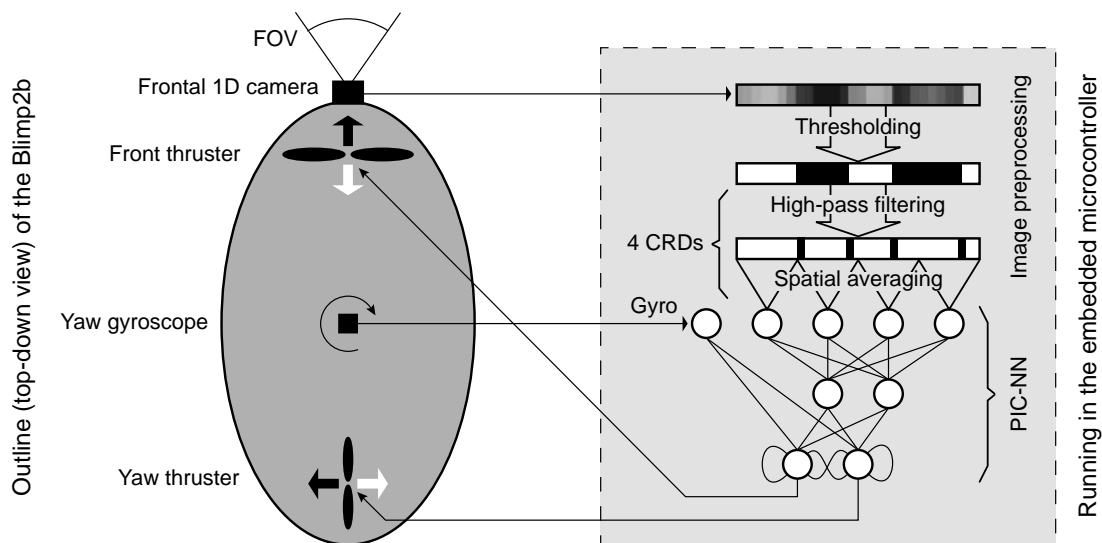


Figure 6.6: Left: Outline of the Blimp2b sensory inputs and actuators. Right: Neural network architecture and vision preprocessing.

yaw gyroscope output is also provided to the neural controller. This additional sensory information is sent to the PIC-NN via an input unit, which is directly connected to the output neurons. One of the motivations for this direct connection is that we know from chapter 5 that a simple proportional feedback loop connecting the gyroscope to the rudder of the airplane is enough to provide course stabilisation.

The PIC-NN has thus 4 visual input units connected to the internal layer, 1 gyroscopic input unit directly connected to the output layer, 2 internal neurons, and 2 output neurons, which control the frontal and yaw thrusters (figure 6.6). The PIC-NN is updated every sensory-motor cycle, which lasts 100ms in reality.<sup>12</sup> The evaluation periods last 1200 sensory-motor cycles (or 2 minutes real-time).

As for the latest experiment with the Khepera robot (subsection 6.2.2), a set of additional precautions are taken during the evolutionary process in order to evolve controllers capable of moving away from walls. The 8 virtual proximity sensors (figure 3.11) are used to set the instantaneous fitness to zero whenever the Blimp2b is less than 25cm from a wall. In addition, individuals that display poor behaviours (less than 0.1 fitness value) are prematurely interrupted after 100 cycles (10 seconds).

<sup>12</sup>A longer sensory-motor cycle than with the Khepera is chosen here, primarily because the communication through the radio system adds some delays. In embedded mode (without monitoring of parameters; see subsection B.2), the sensory-motor cycle could easily be ten times faster.

## Results

Five evolutionary runs were performed, each starting with a different random initialisation. The fitness graph (figure 6.7a) displays steadily increase until the 40th generation. Note that it is far more difficult for the Blimp2b than for the Khepera to approach a fitness value of 1.0 because of inertia and drag effects. However, all five runs produced efficient behaviours in less than 50 generations to navigate in the room in the forward direction while actively avoiding walls. Figure 6.7b illustrates the typical preferred behaviour of the best evolved individuals. The circular trajectory is almost optimal from a velocity point of view, because the circle fits well the available space (the back of the blimp sometimes gets very close to a wall without touching it). Evolved robots do not turn sharply to avoid walls, probably because that would cause a speed loss. The fact that the trajectory is not centered in the room is probably due to the spatial frequency discrepancy between walls (two walls feature less vertical stripes than the two others). The non-zero angle between the heading direction of the blimp (indicated by the small segments) and the trajectory indicates that the simulated flying robot is always side-slipping and thus evolved controllers must take into account the quite complex dynamics by partly relying on air drag to compensate for the centrifugal force.

In order to further assess the wall-avoidance capability of the evolved robots, we artificially reduced the size of the room (another nice feature of the simulation) and tested the same best individual in this new environment. The blimp modified its trajectory into a more elliptic one (figure 6.7c, moving close to the walls without touching them). In another test, when the best individual is deliberately put against a wall (figure 6.7d), it reverses its front thruster, backing away from the wall while rotating in order to recover its preferred circular trajectory. This behaviour typically results from the pressure exerted during evolution by the fact that individuals could be interrupted prematurely if they do not gain some fitness during the first 10 seconds. They are therefore constrained to find an efficient strategy to get out from whatever initial position (even at the expense of a backward movement, which obviously brings no fitness points) in order to quickly resume the preferred high-speed trajectory.

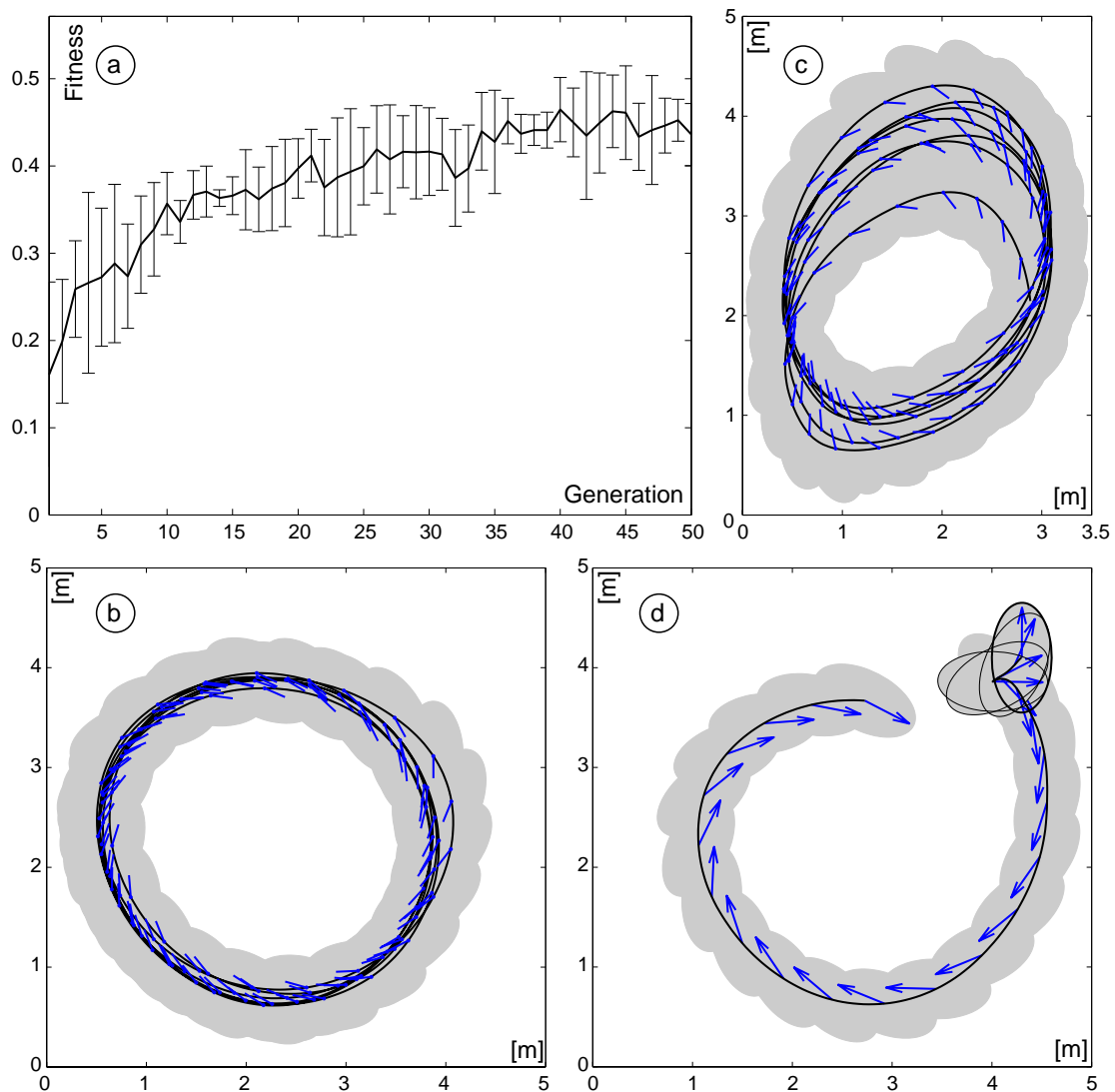


Figure 6.7: Results in simulation. (a) Average fitness values and standard deviations, over a set of five evolutionary runs, of the fittest individuals of each generation. (b) Top-down view of the typical trajectory during 1200 sensory-motor cycles of the fittest evolved individual. The black continuous line is the trajectory plotted with a time resolution of 100ms. The small segments indicate the heading direction every second. Light-gray ellipses represent the envelope of the blimp also plotted every second. (c) Trajectory of the fittest individual when tested for 1200 sensory-motor cycles in a room that has been artificially shrunk by 1.5m. (d) When the same best individual is started against a wall, it first reverses its front thruster while quickly rotating clockwise before resuming its preferred behaviour. The ellipse surrounded by the bold black line indicates the starting position. The following ones with black outline indicate the blimp envelope when the robot is in backward motion. Arrows indicate the longitudinal orientation of the blimp, irrespective of forward or backward movement.

### 6.3.2 Transfer to Reality

When the best evolved neuromorphic controller is tested on the physical robot (without further evolution), it displays almost identical behaviour.<sup>13</sup> Although we cannot measure the exact trajectory of the blimp in reality (the room being not high enough for filming from above), the behaviour displayed by the robot in the 5x5m arena is qualitatively very similar. The Blimp2b is able to quickly drive itself on its preferred circular trajectory, while accurately avoiding contact with the walls.

The fitness function can be used as an estimate of the quality of transfer. A series of comparative tests were performed with the best evolved controller, in simulation and reality. For these tests, the virtual proximity sensors are not used because they do not exist in reality. As a result, the instantaneous fitness is not set to zero when the blimp is close to walls, as it is the case during evolution in simulation. Therefore, the fitness values should be expected to be slightly higher than those shown in the fitness graph of figure 6.7a. The best evolved controller was tested 10 times in simulation and 10 times in reality for 1200 sensory-motor cycles. The results of these test, which are plotted in figure 6.8, show that controllers evolved in simulation obtain very similar performance when assessed on the real robots.

In order to further check the correspondence between simulated and real robot, we compared signals from the anemometer, the gyroscope and the actuators between simulation and reality, while the Blimp2b moves away from a wall. Those signals provide an estimation of the behaviour displayed by the robot. The Blimp2b was thus started facing a wall, as shown in figure 6.7d, both in simulation and in reality. Figure 6.9 shows the very close match between signals gathered in reality and those recorded in the same situation in simulation. At the beginning, the front thruster is almost fully reversed while a strong yaw torque is produced by the yaw thruster. These actions produce the same increment in rotation rate (detected by the gyroscope) and a slight backward velocity (indicated by negative values of the anemometer), both in reality and in simulation. After approximately 3 seconds, the blimp has almost finished the back-and-rotation manoeuvre and starts a strong counter-action with the yaw thruster to cancel the rotational movement, thus resulting in a noticeable decrease of gyroscope output. After that, the robot accelerates forward (as shown in the anemometer

---

<sup>13</sup>Video clips of simulated and physical robots under control of the same evolved neural controller are available at <http://phd.zuff.info>.



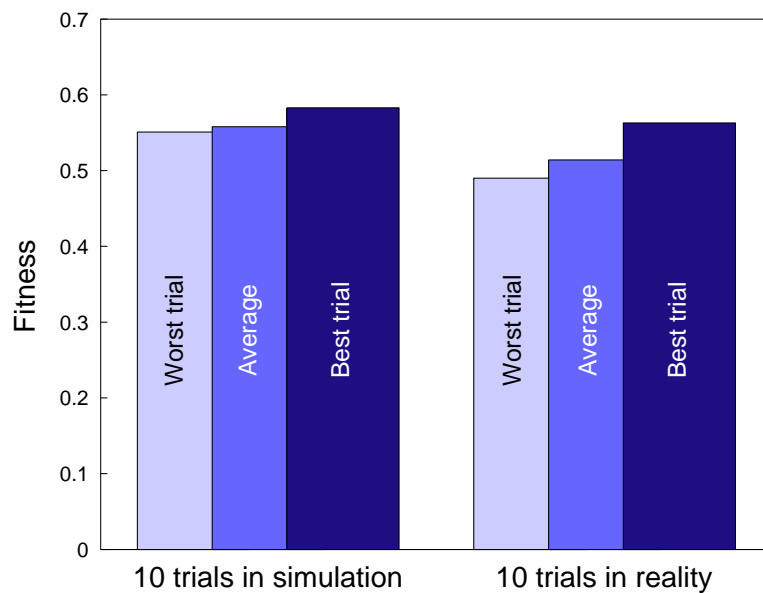


Figure 6.8: Performance from simulation to reality with the same controller. Fitness results of 10 trials of the best evolved individual, left in simulation, right in reality.

graph) to recover its preferred circular trajectory (as revealed by the almost constant, though not null, gyroscope values). Slight discrepancies among signals from simulation and reality can be explained by differences in the starting position implying slightly different visual inputs, inaccuracies in sensor modelling, and omitted higher order components in the dynamic model (appendix A).

## 6.4 Summary and Discussion

In this chapter, we have been interested in exploring alternative strategies to vision-based steering. We decided to use an evolutionary robotics (ER) approach because it can implicitly take care of the constraints related to the robot (sensors, processing power, dynamics), without imposing a specific manner of processing sensory information, nor forcing a pre-defined behaviour for accomplishing the task (maximising forward translation).

Artificial evolution was used to develop a neural controller mapping visual input to actuator commands. With the Khepera robot, evolved individuals display efficient strategies to navigate in the square textured arenas without relying on optic flow. The strategies are based on contrast rate, which is a purely spatial property of the image. When the

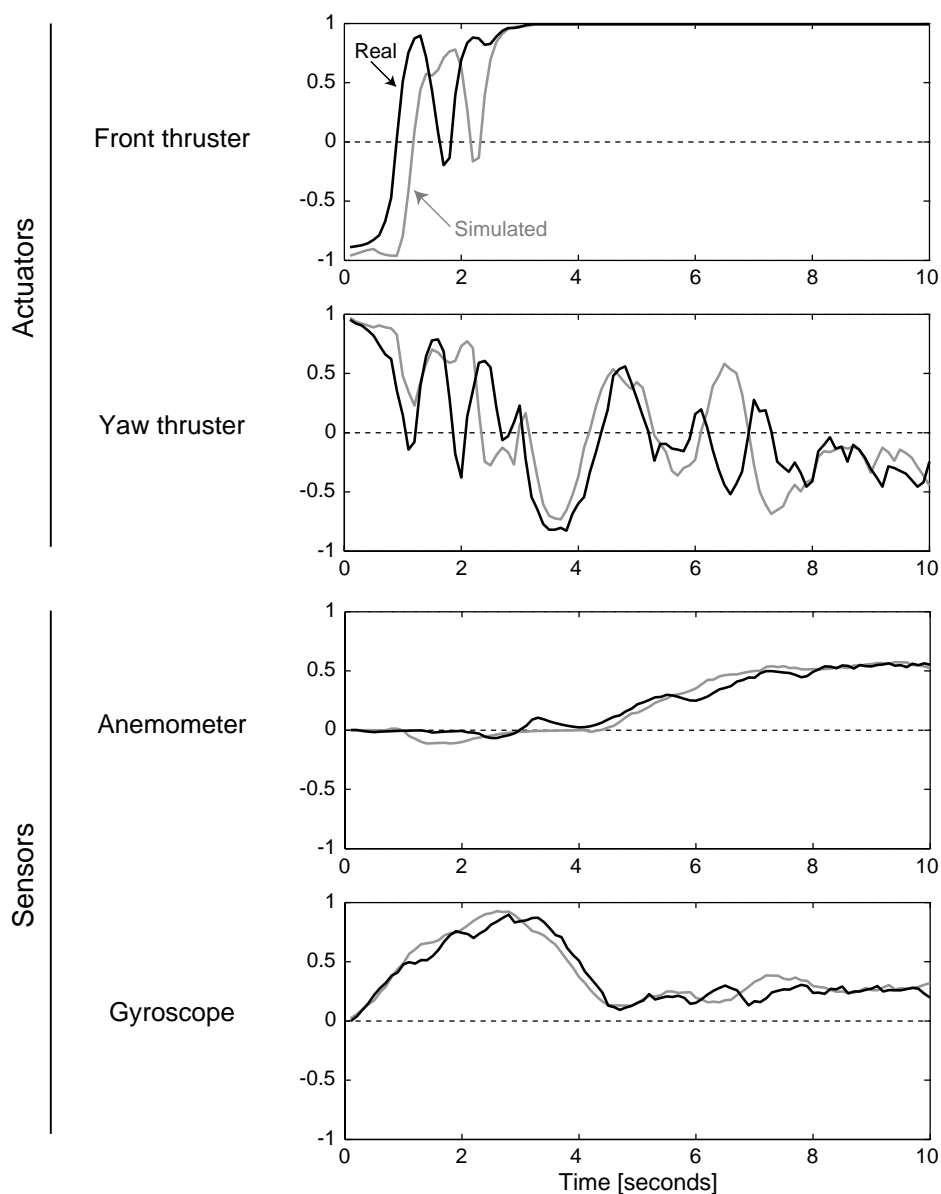


Figure 6.9: Comparison of thruster commands and sensor values between simulation and reality when the best evolved individual is started facing a wall, as shown in figure 6.7.d. Thruster values are normalised with respect to full range; anemometer output is normalised with respect to maximum forward velocity; gyroscope data is normalised with respect to maximum rotation velocity. Note that already after 4 seconds, the robot starts to accumulate fitness since the anemometer is measuring forward motion (during evolution 10 seconds were allowed before interruption due to poor fitness).

same neural controller was explicitly fed with optic-flow, evolution did not manage to find efficient strategies probably because optic-flow requires more complex coordination between motion and perception than

what could be achieved with the simple neural network we employed.

When applied to the Blimp2b, artificial evolution found an efficient way of stabilising the course (CS) and steering the robot so to avoid obstacles (OA), also relying on contrast rate. Additionally, evolved individuals are capable of recovering from critical situations, where it is not possible to simply move forward to get high fitness score.

The evolved neural controllers can operate without any external computer, which is only required during the evolutionary process in order to manage the population of genetic strings.

### **Comparison with Hand-crafting of Bio-inspired Control Systems**

When using ER, the role of the designer is limited to the realisation of the robot (which yield a number of implicit constraints such as dynamics and sensor limitations), the implementation of the controller building blocks (in our case artificial neurons), and the design of a fitness function. The evolutionary process will then try and find the controller configuration that best meets all these constraints. The resulting strategies can be interesting to analyse. In our case, we learnt that image contrast rate is a usable visual cue to drive our robots in the experimental arenas.

However, it is, in some sense, a minimalist solution that will work only under the same conditions as those existing during evolution. In particular, the individuals will fail as soon as the average spatial frequency of the surrounding texture is modified. In contrast, the optic-flow-based solution developed in chapter 5 has been designed to be insensitive to pattern frequency. Also, the asymmetrical evolved behaviour will perform less efficiently in an elongated environment (e.g., a corridor) whereas the symmetrical obstacle avoidance strategy of the F2 airplane will alternatively provoke a saccade to the left and to the right. To tackle those issues, it would be possible to change the environmental properties during evolution, but this would require longer evolutionary runs and probably more complex neural networks.

A significant drawback of ER with respect to hand-crafting bio-inspired controllers is that it requires a large amount of evaluations of randomly initialised controllers. To cope with this issue, either the robot must be capable of supporting such controllers and recovering at the end of every evaluation period, or the use of an accurate, physics-based simulator is inevitable. The development of such a simulator may be quite difficult depending on the dynamics of the robot, the

complexity of the environment, and the type of sensors employed.<sup>14</sup>

### **Evolutionary Approach and Fixed-wing Aircraft**

Airplanes such as the F2 would not support an evolutionary run performed on the physical robot for three reasons. First, it is not robust enough to withstand repeated collisions with the walls of the arena. Second it cannot automatically initialise itself in a good (airborne) posture at the beginning of every evaluation period. Third, it has a very limited energetic autonomy (about 20-30 minutes). The only solution to apply the evolutionary approach to such an airplane is to develop an accurate flight simulator. However, this is more difficult than with an airship because under the control of a randomly initialised neural controller the airplane will not only fly in its standard regime (near level flight at reasonable speed), but also in stall situations, or high pitch and roll angles. Such non-standard flight regimes are difficult to model since unsteady-state aerodynamics plays a predominant role.

To cope with this issue, some precautions can be envisaged. For instance, it is conceivable to initialise the robot in level flight with its nominal velocity and prematurely interrupt the evaluation period whenever some parameters (such as pitch and roll angles, and velocity) leave a predefined range where the simulation is known to be accurate. This will also force the individuals to fly the plane in a reasonable regime.

Other problems related to simulation-reality discrepancies could be approached with other techniques. Incremental evolution consisting in pursuing evolution in reality for a short amount of generations (see Harvey et al., 1994 or Nolfi and Floreano, 2000, section 4.4) could be a first solution, although a human pilot would probably be required to initialise the aircraft and rescue it whenever the controller fails. Moreover, the procedure could be very time-consuming and risky for the robot. The second approach consists in using some sort of synaptic plasticity in the neural controller. Exploitation of synaptic adaptation has been shown to support fast self-adaptation to changing environments (Urzelai and Floreano, 2001).

---

<sup>14</sup>See Nolfi and Floreano (2000) for a detailed discussion about the use of simulation in ER.

**Outlook**

In this thesis, artificial evolution was used exclusively to set the synaptic strength of a simple neural network. However, artificial evolution in simulation could be used to explore architectural issues such as airframe shape (provided that the simulator is able to infer the effects on the dynamics) or sensor morphology (Cliff and Miller, 1996; Huber et al., 1996; Lichtensteiger and Eggenberger, 1999). For instance, position and orientation of simple vision sensors could be left to evolutionary control and the fitness function could put some pressure toward the use of a minimum number of sensors. Ultimately, artificial evolution could also allow to explore higher order combinations of behaviours (taking-off, flying, avoiding obstacles, going through small apertures, looking for food, escaping predators, landing, etc.), for which we still know so little in insects.



# Chapter 7

## Concluding Remarks

I see insect level behavior as a noble goal for artificial intelligence practitioners. I believe it is closer to the ultimate right track than are the higher level goals now being pursued.

---

R.A. Brooks (1986)

### 7.1 Main Achievements

Inspired by flying insects, we demonstrated autonomous steering of a 30-gram indoor airplane using only visual and gyroscopic sensors. The signal processing and control are entirely computed on-board the plane, despite its very limited payload of approximately 6g. This application forced us to develop ultra-light optic-flow detectors and to fit the algorithms (optic flow detection and airplane control) in a tiny 8-bit microcontroller running at 20MHz. The entire software running on the airplane uses less than 20kB of program memory. In flight, the airplane consumes less than 2W, which is 30 times less than a desk light or 50 times less than a small hobby helicopter. Our other flying platform, the blimp, allowed us to use an evolutionary technique in order to develop vision-based neural controllers also running in the embedded microcontroller. The blimp requires only 1W to autonomously circumnavigate the experimental arena while avoiding collisions.

One of the main outcomes of this thesis is the insight gained on how to link simple visual features (such as local optic-flow patterns or contrast rate) to motor commands in order to obtain efficient behaviours with lightweight and dynamic robots featuring limited computational

resources. Some of the problems arising when optic-flow is used in absence of contact with an inertial frame (no odometry, unstable motion) have been solved by merging gyroscopic information to vision. The results of the evolutionary experiments showed that optic flow is not the only way of processing monocular visual information for course control and obstacle avoidance. Although the evolved contrast-rate solution cannot be generalised to other environments as easily as an optic-flow-based strategy, it represents an efficient alternative, which requires even less computational power.

Although the primary purpose of this project was to engineer indoor autonomous flyers, the size, energy, and computational constraints of the robotic platforms encouraged us to look at mechanisms and principles of flight control exploited by insects. Our approach to developing autonomous vision-based flying robots has been inspired by biology at different levels: low-resolution insect-like vision, information processing, behaviours, neural networks, and artificial evolution. In doing so, we have, in some sense, contributed to test different biological models, in particular with the demonstration that an artificial "flying insect" could steer autonomously over a relatively long period of time. In this regard, this thesis is an illustration of the synergistic relationship that can exist between robotics and biology.

## 7.2 Potential Applications

This work can be of great help in automating very light flying devices such as those presented in subsection 1.2.1, which feature the same kind of properties as our flying platforms: complex dynamics due to the absence of contact with an inertial frame, limited payload and restricted computational power. Distance sensors are not a viable solution in such cases, and visual and gyroscopic sensors are probably the best alternative to provide such robots with basic navigational skills. This seems to be acknowledged by the micromechanical flying insect (MFI) team in Berkeley, who is currently working on a sensor suite for their 25mm flapping robot (figure 1.2). Although obstacle avoidance has not yet been tackled, preliminary work toward attitude control relies on visual (ocelli-like) and gyroscopic (halteres-like) sensors (Wu et al., 2003; Schenato et al., 2004).

More generally, the approach proposed in this thesis can provide low-level navigation strategies for all kinds of mobile robots featuring



small size and non-trivial dynamics. It can equally be useful in a number of situations where the environment is unknown (no precise maps are available) and the use of GPS is not possible, which is the case for example in indoor environments, underwater, or in planetary exploration, especially if the robot has to move close to the relief or in cluttered environments that are difficult to reconstruct with range finders. This could also be useful for UAVs or MAVs flying at low altitude in urban or natural environments, where buildings, trees, hills, etc. may be present and a fast collision avoidance system is required (Mueller, 2001).

Beyond the application of vision-based navigation strategies to mobile robots, the use of small indoor flying systems as biological tools can be envisaged. As indicated in the introduction, more and more biologists are assessing their models with mobile robots. However, until now, the used robotic platforms were only terrestrial vehicles (e.g., Srinivasan et al., 1998) or tethered systems (e.g., Reiser and Dickinson, 2003; Ruffier, 2004). An indoor flying platform with visual sensors and the ability to fly at velocities close to the one reached by flies (1-3m/s) potentially provides a great testbed for assessing models of visually guided behaviours in free flight. The fact that our aerial robots are small, resistant to crash and fly indoor, further eases the testing phase and alleviates the need of large technical teams.

### **7.3 Future Directions**

Unlike tethered or simulated aircrafts, free-flying physical systems cannot readily know their position and orientation, which is highly desirable for analysis purposes during the testing phase. In the future, some efforts should be put in the development of an off-board positioning system functioning without any heavy or energy consuming parts on the robots. It is probable that a tracking system relying on several ground-based cameras represents a good option. Such a system would also be of great help in assessing and fine tuning a flight simulator. It would allow to compare trajectories obtained in reality to those obtained in simulation when the the same actuator commands are issued to the airplane.

Another issue that must be tackled is autonomous flight in unconstrained indoor environment. In order to reach that goal, more attention will be needed at the level of the visual sensors. Two challenges

have to be addressed. First, the intensity changes among different rooms (some receiving direct sun light and others having only artificial lighting) might be enormous. Second, it may happen that parts of the visual surroundings have absolutely no contrast. If the vision system samples the field of view only in few areas (like it is currently the case with our robots) it will provide no usable signals in those situations.

To cope with those issues while maintaining an overall low weight and low power consumption is a challenge that might be addressed with aVLSI<sup>1</sup> neuromorphic visual sensors (Liu et al., 2003) in place of using classical CMOS cameras. The neuromorphic technology provides a circuit-design approach to the possibility of implementing certain natural computations more efficiently than standard logic circuits. The resulting chips usually consumes at least 10 times less power than an equivalent implementation with CMOS imagers and digital processor. More specifically appealing for tackling the problem of background light fluctuations is the existence of adaptive photoreceptor circuits (Delbrück and Mead, 1995) that automatically adapt to background light over very large intensity range (more than 6 decades). Those photoreceptors can be used as front-end to optic-flow detector circuits fitted on the same chip (e.g., Kramer et al., 1995). This technology also provides the potential for widening the field of view in arranging pixels and optic-flow circuits as desired on a single chip, while consuming less energy and computational power than if the same functionality had to be achieved with standard CMOS sensors and vision processing in a microcontroller. As an example, Harrison (2003) presented an aVLSI chip for imminent collision detection based on the STIM model (subsection 2.3.3).

Finally, the science fiction of an artificial flying insect buzzing around your office, suddenly deciding to escape through the door and managing to reach your colleague's room is a futuristic scenario that this thesis humbly contributes to bring closer to reality.

---

<sup>1</sup>Analog Very Large Scale Integration.

# Appendix A

## Dynamic Modelling of Indoor Airships

In this appendix, a generic dynamic model for indoor airships (blimps) is presented together with a pragmatic methodology for parameter identification. This theoretical work<sup>1</sup> has been applied to the Blimp2b (subsection 3.1.2) to provide our robotic simulator (subsection 3.3.2) with an accurate dynamic modelling featuring good correspondence between reality and simulation (see, e.g., subsection 6.3.2).

The dynamic modelling is generic to indoor airships with a hull shape that can be approximated by an ellipsoid of revolution. The steering of the airship is assumed to be done by means of thrusters (e.g., a DC motor with a propeller), whose number, orientation and location can be freely chosen. The typical maximum velocity of such aerial vehicles is around 1m/s and their length is usually in the range of 1 to 5m. The hull and gondola distortions are assumed to be small and to have negligible impact on trajectory, allowing to approximate the whole airship as a rigid body. The airship has two vertical planes of symmetry, with the center of gravity (CG) and the center of buoyancy (CB) located in their intersection. These assumptions generally apply well to such kinds of small airships with inflated envelope and light-weight gondola. Fortunately, they generate significant simplifications in the equation of motion and allow easier parameter identification.

In order to describe the motion of the airship in the 6 degrees of freedom (DOF) and the forces and moments acting on it, we define two reference frames: an earth inertial reference frame  $\mathcal{F}_E$  and a body-fixed frame  $\mathcal{F}_B$  (figure A.1). The origin of  $\mathcal{F}_E$  is an arbitrary point on

---

<sup>1</sup>This work has been carried out together with Alexis Guanella during his diploma project (Guanella, 2004).

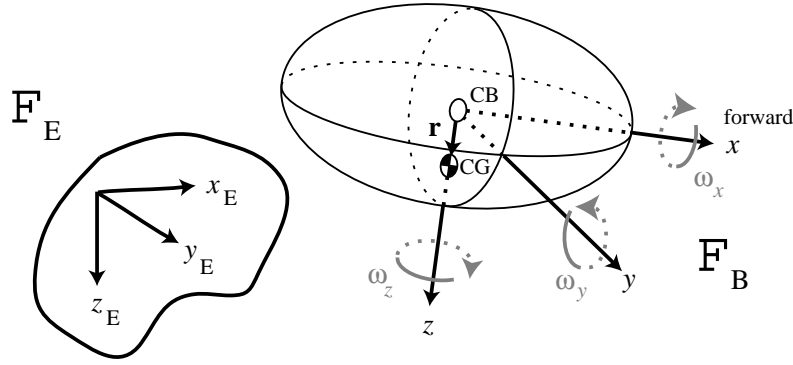


Figure A.1: Inertial frame and body-fixed frame.

Earth's surface, whose acceleration is neglected. Note that the  $z_E$ -axis is pointing downward. The origin of the body-fixed frame  $\mathcal{F}_B$  is the center of buoyancy CB, which corresponds to the center of the hull. Its orientation coincides with the orientation of the airship and its axes correspond to principal axes of the rigid body. The center of gravity CG is located by the vector  $\mathbf{r} = (0, 0, r_z)^T$  in  $\mathcal{F}_B$ .

Translational and rotational velocities  $\boldsymbol{\nu} := (\mathbf{v}^T, \boldsymbol{\omega}^T)^T = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T$  are described in the body-fixed frame  $\mathcal{F}_B$ , whereas the position and orientation of the vehicle  $\boldsymbol{\eta} := (x_E, y_E, z_E, \phi, \theta, \psi)^T$  are expressed in the inertial frame  $\mathcal{F}_E$ . Note that the last 3 terms of  $\boldsymbol{\eta}$  are the aeronautical Euler angles, namely roll  $\phi$ , pitch  $\theta$ , and yaw  $\psi$ .

The Newton-Euler equation of motion links the acceleration of the airship with the forces and moments acting on it. This non-linear equation is written using a vector representation for 6 DOF in the body-fixed frame  $\mathcal{F}_B$ :

$$\mathbf{M}\dot{\boldsymbol{\nu}} = \sum \mathbf{F}_{external} = \mathbf{F}_R + \mathbf{F}_P + \mathbf{F}_D + \mathbf{F}_C, \quad (\text{A.1})$$

where the five main elements are listed below, following the order in which they will be presented:

- $\mathbf{F}_R$ : restoring forces (section A.1) containing gravity and buoyancy, which counteract each other and are responsible for maintaining the airship upside up;
- $\mathbf{F}_P$ : propelling forces (section A.2), which are directly related to motor commands;
- $\mathbf{F}_D$ : damping forces (section A.3) due to air friction;

- M: inertia matrix (section A.4) containing rigid-body inertia and added mass terms;
- $F_C$ : Coriolis and centripetal effects (section A.5), which are fictitious forces appearing in non-inertial frames such as  $\mathcal{F}_B$ .

The presentation order of those elements is motivated by both the identification process, which sometimes requires the value of previous components to be determined first (e.g. propelling thrusts are needed for the measurement of damping forces), and the fact that we use the inertia matrix as a basis for the derivation of the Coriolis matrix (section A.5).

## A.1 Restoring Forces

Airships counteract gravity by some lift capability. Unlike airplanes, the aerostatic lift force (buoyancy) acting on an airship is independent of flight speed. The buoyant force is explained by Archimedes' principle and is equal to the mass of the volume of air displaced by the airship. Gravity and buoyancy together are called restoring forces since they are responsible for keeping the airship upside up. The amplitudes of these forces are expressed by:

$$F_g = mg \quad \text{and} \quad F_b = \rho V g, \quad \text{with} \quad V = \frac{4}{3}\pi ab^2, \quad (\text{A.2})$$

where  $g$  is the Earth's gravitational acceleration,  $\rho$  is the air density, and  $V$  the volume of the hull, which is an ellipsoid of revolution with semi-axes  $a$ ,  $b$  and  $b$  (with  $a > b$ ). Both restoring forces are always parallel to the  $z_E$ -axis and the yaw angle  $\psi$  does not appear in the modelling of this phenomenon, which is expressed in the body-fixed frame  $\mathcal{F}_B$  in the following form:

$$\mathbf{F}_R(\boldsymbol{\eta}) = \begin{pmatrix} -(F_g - F_b) \sin(\theta) \\ (F_g - F_b) \cos(\theta) \sin(\phi) \\ (F_g - F_b) \cos(\theta) \cos(\phi) \\ -r_z F_g \cos(\theta) \sin(\phi) \\ -r_z F_g \sin(\theta) \\ 0 \end{pmatrix}. \quad (\text{A.3})$$

The only non-trivial parameter to identify at that point is  $r_z$ , namely the distance between CB and CG. In order to measure  $r_z$ , instead of

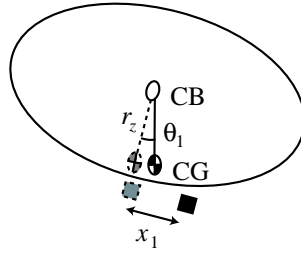


Figure A.2: Procedure for the localisation of the center of gravity.

computing it based on the repartition of every masses on the body, we rather propose to temporarily modify the mass distribution along the  $x$ -axis while the airship is airborne and to measure the resulting static pitching angle. It is often easy to do so by displacing a part of the ballast, say with a mass of  $m_1$ , along the  $x$ -axis, at a distance  $x_1$  from CB (figure A.2). Then the resulting pitching angle  $\theta_1$  allows for deriving  $r_z$  using the following formula based on simple geometric considerations:

$$r_z = \frac{m_1}{m} \frac{x_1}{\tan(\theta_1)}. \quad (\text{A.4})$$

## A.2 Propulsion

In our model, engines are assumed to be ideal thrusters applying forces, whose amplitudes are specified by the motor commands, and neither propeller fluxes nor motor torques are taken into account. Therefore, the propulsion matrix  $F_P$  depends only on motor commands and the location and orientation of the engines. Since the thrusters are moving with  $\mathcal{F}_B$ , it is straight forward to express the resulting forces and torque in the body-fixed frame. The identification procedure is thus limited to the determination of the thrust amplitude as a function of motor commands. To that end, we simply propose to measure thrust for each engine. The outcome of this experiment will then be used to construct a look-up table where corresponding forces will be found whenever a motor command is issued.

## A.3 Damping

Aerodynamic damping is due to air friction, which is a function of velocity. In general, there are two different regimes that can be distin-

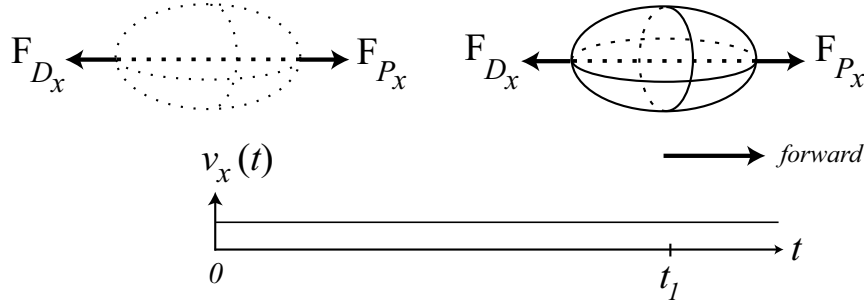


Figure A.3: Identification of damping coefficients.

guished: linear friction due to laminar boundary layers and quadratic friction due to turbulent boundary layers. Since it is difficult to know in advance in which regime the airship is operating, we model the damping as a Taylor series where the terms higher than second order are neglected.

$$\mathbf{F}_D = \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}, \text{ with } \mathbf{D}(\boldsymbol{\nu}) = -diag \begin{pmatrix} D_{v_x} + D_{v_x^2} |v_x| \\ D_{v_y} + D_{v_y^2} |v_y| \\ D_{v_z} + D_{v_z^2} |v_z| \\ D_{\omega_x} + D_{\omega_x^2} |\omega_x| \\ D_{\omega_y} + D_{\omega_y^2} |\omega_y| \\ D_{\omega_z} + D_{\omega_z^2} |\omega_z| \end{pmatrix}, \quad (\text{A.5})$$

where  $\mathbf{D}(\boldsymbol{\nu})$  is the damping matrix,  $D_{v_x}$ ,  $D_{v_y}$ ,  $D_{v_z}$ ,  $D_{\omega_x}$ ,  $D_{\omega_y}$ ,  $D_{\omega_z}$  are the linear damping coefficients, and  $D_{v_x^2}$ ,  $D_{v_y^2}$ ,  $D_{v_z^2}$ ,  $D_{\omega_x^2}$ ,  $D_{\omega_y^2}$ ,  $D_{\omega_z^2}$  the quadratic damping coefficients. Note that this uncoupled model of damping is a rough approximation that works sufficiently well in the case of low speed and highly symmetrical ellipsoid hull (Fossen, 1995).

It is possible to identify the 12 damping coefficients by measuring stationary velocities reached by the airship subjected to different constant forces. For instance, a known thrust  $F_{P_x}$  is applied in the forward direction by means of one or several engines with constant thrust (figure A.3). When the blimp reaches a constant forward velocity, inertial and Coriolis effects are null and the following extract of the equation of motion can be used to reveal the two damping coefficients related to the x-axis:

$$F_{P_x} - D_{v_x} v_x - D_{v_x^2} v_x^2 = 0, \text{ with } v_x > 0. \quad (\text{A.6})$$

This experiment must be run for different values of  $F_{P_x}$ , which can

be known from section A.2 and the related constant velocities  $v_x$  can be measured using a stopwatch while the airship is travelling along a known distance. Then the linear and quadratic damping coefficients are determined such that the curve  $v_x(F_{P_x})$  defined by equation (A.6) best fits the experimental data.

The same method is used for the  $y$  and  $z$ -axes. In some cases however, the airship might not be equipped with engines for lateral motion. As a consequence, this method cannot be applied and lateral damping coefficients should rather be approximated from the vertical ones. This is possible because of the symmetric hull:  $D_{v_y} \simeq D_{v_z}$  and  $D_{v_y^2} \simeq D_{v_z^2}$ . Concerning rotational coefficients, the same approach can be followed, at least for  $D_{\omega_z}$  and  $D_{\omega_z^2}$ . The airship must be accelerated around the yaw axis using a constant thrust until it reaches a stationary velocity, which can be measured by counting the number of revolutions per time unit.

Since airships are rarely designed to make full rolling or pitching movements, the same procedure cannot be applied to pitch and roll damping coefficients. Even if such movements were possible, the restoring effect would prevent a simple modelling like in equation (A.6). Fortunately, for the pitch axis, the same considerations as above hold and we have  $D_{\omega_y} \simeq D_{\omega_z}$  and  $D_{\omega_y^2} \simeq D_{\omega_z^2}$ . For the roll axis, the damping coefficients  $D_{\omega_x}$  and  $D_{\omega_x^2}$  are definitely very close to zero since the cross section is comparable to a circle. However, if they are put to zero, the simulated airship will always oscillate about this axis. A crude estimation would be to use values around half the ones obtained for the yaw axis<sup>2</sup>.

## A.4 Inertia

Using the same notations as in (Fossen, 1995), the rigid-body inertia matrix can readily be written as follows:

---

<sup>2</sup>If damping in rolling and pitching behaviours are important for a particular experiment (which is not our case), one could envisage a more complex identification procedure that would consist in letting the airship oscillate around those axes (similar to the experiment proposed in the following section) and analyse the movement, e.g., with a laser mounted in the gondola and pointing downward to the lab floor, but this is out of the scope of this appendix.



$$\mathbf{M}_{RB} = \begin{pmatrix} m\mathbf{I}_{3 \times 3} & -m\mathbf{S}(\mathbf{r}) \\ m\mathbf{S}(\mathbf{r}) & \mathbf{I}_{RB} \end{pmatrix}, \quad \text{with } \mathbf{S}(\mathbf{a}) := \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}, \quad (\text{A.7})$$

where  $m$  is the mass of the airship,  $\mathbf{I}_{3 \times 3}$  is the identity matrix,  $\mathbf{S}$  is the skew-symmetric matrix operator ( $\mathbf{a} \in \mathbb{R}^3$ ) and  $\mathbf{I}_{RB}$  is the inertia tensor with respect to CB. Taking into account that the axes of  $\mathcal{F}_B$  are the principal axes yields a diagonal inertia tensor. Further simplification can be made by recalling that, in our case,  $\mathbf{r} = (0, 0, r_z)^T$ . Finally, explicit description of the rigid-body inertia matrix yields:

$$\mathbf{M}_{RB} = \begin{pmatrix} m & 0 & 0 & 0 & mr_z & 0 \\ 0 & m & 0 & -mr_z & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & -mr_z & 0 & I_x & 0 & 0 \\ mr_z & 0 & 0 & 0 & I_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z \end{pmatrix}. \quad (\text{A.8})$$

This representation of inertia is however not sufficient, since a bulky body in motion displaces quite a number of particles of the surrounding fluid in its direction of movement. This phenomenon has a noticeable impact on buoyant vehicles, which have by definition similar density as their surrounding fluid. Therefore, the body experiences a resistance to its motion, which is not accounted for by the standard rigid-body inertia matrix described above. This additional effect is modelled by including added mass and inertia terms into both inertia and Coriolis matrices. In short, "added-mass" (sometimes also called "virtual mass") is a measure of the additional inertia created by surrounding air accompanying the airship. As in Fossen (1995), we propose a simple modelling of the added-mass effect by introducing a diagonal added-mass inertia matrix<sup>3</sup>:

$$\mathbf{M}_A = \text{diag}(m_{A_x}, m_{A_y}, m_{A_z}, I_{A_x}, I_{A_y}, I_{A_z}). \quad (\text{A.9})$$

Finally, we derive the global inertia matrix  $\mathbf{M}$  as the sum of rigid-

<sup>3</sup>This is tenable in our case since indoor blimps are moving at very low speed and we assume three planes of symmetry for the ellipsoidal hull, which is essentially responsible for this effect, allowing to neglect the gondola added-mass. This reduction is also motivated by the fact that off-diagonal elements are difficult to determine from experiments as well as theory Fossen (1995). Note however that in general, for higher speed and more complex shapes, coupling terms should be taken into account.

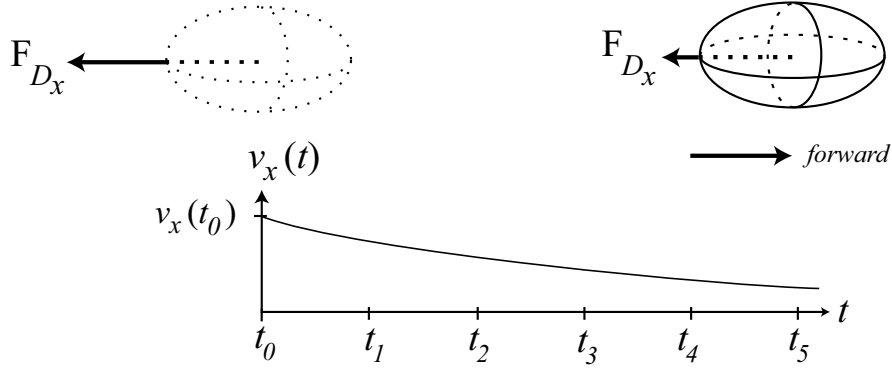


Figure A.4: Identification of apparent translational mass.

body inertia and added-mass matrices:

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A = \begin{pmatrix} m'_x & 0 & 0 & 0 & mr_z & 0 \\ 0 & m'_y & 0 & -mr_z & 0 & 0 \\ 0 & 0 & m'_z & 0 & 0 & 0 \\ 0 & -mr_z & 0 & I'_x & 0 & 0 \\ mr_z & 0 & 0 & 0 & I'_y & 0 \\ 0 & 0 & 0 & 0 & 0 & I'_z \end{pmatrix}, \quad (\text{A.10})$$

where  $m'_x := m + m_{A_x}$ ,  $m'_y := m + m_{A_y}$ ,  $m'_z := m + m_{A_z}$ ,  $I'_x := I_x + I_{A_x}$ ,  $I'_y := I_y + I_{A_y}$ , and  $I'_z := I_z + I_{A_z}$  are respectively the apparent masses and moments. Note that the shape of the envelope readily suggests that  $m_{A_x} < m_{A_y} \approx m_{A_z}$ ,  $I_{A_x} \approx 0$  and  $I_{A_y} \approx I_{A_z}$  (Munk, 1934). At this point, it could be tempting to neglect the added mass and inertia, but one should be aware that this phenomenon is responsible for an intrinsic instability of airships (see section A.5) and omitting it would hardly lead to realistic behaviour in simulation.

The identification procedure for  $\mathbf{M}$  concerns only the 6 diagonal elements, since the static mass is known and the distance  $r_z$  has been determined in section A.1. There are essentially two ways of tackling the task and both will be described. A first approach, following the philosophy of our approach, would be to directly measure the apparent masses and moments as a whole, i.e., without dissecting them into  $\mathbf{M}_{RB}$  and  $\mathbf{M}_A$  terms. Unlike for the damping coefficients, in this case, we need acceleration in order to reveal inertia effects, which would be null otherwise. The experiment consists therefore in setting the blimp at a known velocity along a particular axis, then cutting off engines and

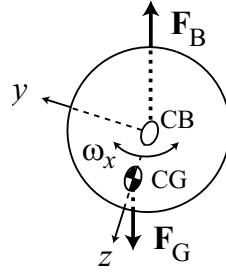


Figure A.5: Identification of apparent momentum of inertia.

measuring the time it takes to travel along several chunks of known distance while it decelerates (figure A.4). The only effects acting on the airship during this free movement are inertia (which we want to measure) and damping (which is already known from section A.3). The equation describing this particular motion along, e.g., the  $x$ -axis can be extracted from equation (A.1):

$$m'_x \dot{v}_x = F_{D_x} = -D_{v_x} v_x - D_{v_x^2} v_x^2, \text{ with } v_x \geq 0. \quad (\text{A.11})$$

This differential equation yields the following solution, where  $v_0$  is the initial speed:

$$v_x(t) = -\frac{D_{v_x} e^{\frac{D_{v_x} t}{m'_x}} v_0}{D_{v_x} + D_{v_x^2} (1 - e^{\frac{D_{v_x} t}{m'_x}}) v_0}, \quad (\text{A.12})$$

which we can be integrated to find the position over time:

$$x(t) = \frac{m'_x \log\left(\frac{D_{v_x} + D_{v_x^2} v_0 - D_{v_x^2} e^{\frac{D_{v_x} t}{m'_x}} v_0}{D_{v_x}}\right)}{D_{v_x^2}}. \quad (\text{A.13})$$

$m'_x$  can then be numerically estimated so that  $x(t)$  best fits the points recorded during experiments. The same procedure can be applied for  $m'_y$ ,  $m'_z$  (note that  $m'_y \approx m'_z$ ), and  $I'_z$ , using equation (A.13) adapted to the respective axis.

Rolling and pitching inertia are not identifiable that way, since full rotations around those axes are not feasible. Fortunately, another experiment exists for those axes, which consists in letting the blimp oscillate like a pendulum and measuring its eigen frequency (figure A.5). For this experiment, we neglect damping effects and assume zero velocity and acceleration for all axes excepted for the one we want to measure the inertia. The equation describing this oscillating movement can be

inferred from (A.1). Here is the result for the roll axis:

$$I'_x \dot{\omega}_x + r_z F_g \sin(\phi) = 0. \quad (\text{A.14})$$

Since the blimp is assumed to be steady except for the rotation around its roll axis, we can set  $\dot{\omega}_x = \ddot{\phi}$ . Furthermore, assuming small roll angles around equilibrium permits to approximate  $\sin(\phi)$  by  $\phi$ . This allows to rewrite equation (A.14) in the form of the well-known equation of the conservative oscillator with its eigen frequency denoted  $f_0$  and angular pulsation  $\omega_0$ :

$$\ddot{\phi} + \omega_0^2 \phi = 0, \quad \text{with} \quad \omega_0 := \sqrt{\frac{r_z F_g}{I'_x}} \quad \text{and} \quad f_0 := \frac{1}{2\pi} \omega_0. \quad (\text{A.15})$$

The experimental procedure thus consists in destabilising the airship with a small roll angle and deriving  $f_0$  by simply counting the number of oscillations per time unit. Once the eigen frequency is known,  $I'_x$  can be worked out from equation (A.15), with  $r_z$  and  $F_g$  given in section A.1. The same procedure holds for  $I'_y$ .

Although this method works very well for rolling and pitching moments of inertia, it has some drawbacks, especially for what concerns its first part. On the one hand, the decelerating translational or rotational movements of the airship are difficult to measure with precision. On the other hand, equation (A.13) is quite heavy to handle and has no analytical solution. Therefore, we propose an alternative method relying more on theory than on experiments.

The second approach first proceeds by separating  $M_{RB}$  and  $M_A$  terms. Since the distribution of masses on the airship is known, the inertia tensor can be calculated explicitly (be aware not to forget the inertia of the helium mass in the hull) and from it the full  $M_{RB}$  matrix can be derived. Added-mass factors populating  $M_A$  can then be estimated from a geometrical method based on the kinetic energy of an ideal unbounded liquid around the ellipsoid in motion (here we assume once again that the whole solid can be approximated by an ellipsoid of revolution). The kinetic energy and the force necessary to accelerate the airship can be computed by adding to the actual mass of the solid a fictitious mass. This added-mass is equal to the density of the fluid multiplied by a volume, which depends on the geometric outlines of the airship only (Munk, 1934). These considerations result in the Lamb's  $k$ -factors (Lamb, 1932), where  $k_1$  and  $k_2$  are the inertia coefficients representing the fraction of the mass displaced by the hull (which is in turn

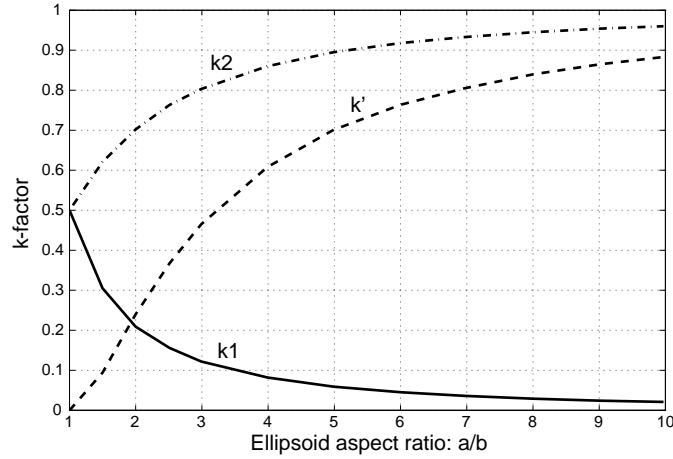


Figure A.6: Lamb factors as a function of the ellipsoidal hull aspect ratio (adapted from (Munk, 1936)). With  $k_1$  the axial coefficient ( $x$ -axis),  $k_2$  the lateral coefficient ( $y$ -axis) and  $k'$  the rotational one (for yaw and pitch axes).

equal to the physical mass  $m$  of the airship if we assume that gravity and buoyancy are balancing each other, see section A.1), whereas  $k'$  is the ratio of the apparent moment of inertia to the moment of inertia of the displaced air  $I_{z_h}$ . In the case of an ellipsoid of revolution with semi-axes  $a$  and  $b$  (with  $a \geq b$ ), this moment of inertia is given by:

$$I_{z_h} = \frac{4}{15} \pi \rho a b^2 (a^2 + b^2). \quad (\text{A.16})$$

The added-mass terms can then be calculated using the Lamb's  $k$ -factors as follows:

$$\begin{aligned} m_{A_x} &= k_1 m \quad \text{and} \quad m_{A_y} = m_{A_z} = k_2 m, \\ I_{A_x} &= 0 \quad \text{and} \quad I_{A_y} = I_{A_z} = k' I_{z_h}. \end{aligned} \quad (\text{A.17})$$

The Lamb's  $k$ -factors are in turn defined using two constants  $\alpha_0$  and  $\beta_0$ :

$$\begin{aligned} k_1 &= \frac{\alpha_0}{2 - \alpha_0}, \quad \alpha_0 = \frac{2(1-e^2)}{e^3} \left( \frac{1}{2} \ln \frac{1+e}{1-e} - e \right), \\ k_2 &= \frac{\beta_0}{2 - \beta_0}, \quad \beta_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3} \ln \frac{1+e}{1-e}, \\ k' &= \frac{e^4 (\beta_0 - \alpha_0)}{(2-e^2)[2e^2 - (2-e^2)(\beta_0 - \alpha_0)]}. \end{aligned} \quad (\text{A.18})$$

where  $e$  designates the eccentricity of the ellipsoid:

$$e = \sqrt{1 - \left( \frac{b}{a} \right)^2}. \quad (\text{A.19})$$

Figure A.6 displays these  $k$ -factors as a function of the ellipsoid as-

pect ratio, starting from the spherical case ( $a = b$ ), ending with a very slender hull having the long axis up to ten times the radius ( $a = 10b$ ). Interestingly, a spherical hull has already 50% added-mass in all directions and no additional moment of inertia ( $k' = 0$ ). With the increase in aspect ratio, the axial added-mass ( $x$ -axis) tend to decrease, whereas it augments in the lateral direction ( $y$ -axis). Note that alternative representation of added-mass terms can be found in (Fossen, 1995; Khoury and Gillet, 1999) and are equivalent to the one originally proposed by Lamb. Further development for generic ellipsoids is also present in (Lamb, 1932; Munk, 1934).

## A.5 Coriolis and Centripetal Effects

Coriolis<sup>4</sup> and centripetal effects are fictitious forces exerted on a body in motion when the referential frame is not inertial, which is generally the case for body-fixed frames. The Coriolis force is expressed as  $\boldsymbol{\omega} \times \mathbf{v}$  and is only apparent when the motion is composed of linear and rotational velocities. It explains the apparent force acting perpendicular to the linear velocity vector and rotation axis, and tends to maintain the initial direction of motion without taking care of the body rotation. The centripetal force is given by  $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})$  and is present in a rotating body when the origin of the referential frame is not CG, which is the case for  $\mathcal{F}_B$ . Since those fictitious forces are similar and both function of  $\boldsymbol{\nu}$ , they are generally put together in this form:

$$\mathbf{F}_C = \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu}, \quad (\text{A.20})$$

where  $\mathbf{C}(\boldsymbol{\nu})$  is the so-called Coriolis matrix. After (Sagatun and Fossen, 1991), it is possible to directly derive the Coriolis matrix from the inertia matrix as follows:

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{pmatrix} \mathbf{O}_{3 \times 3} & \mathbf{S}(\mathbf{M}_{11}\mathbf{v} + \mathbf{M}_{12}\boldsymbol{\omega}) \\ \mathbf{S}(\mathbf{M}_{11}\mathbf{v} + \mathbf{M}_{12}\boldsymbol{\omega}) & \mathbf{S}(\mathbf{M}_{21}\mathbf{v} + \mathbf{M}_{22}\boldsymbol{\omega}) \end{pmatrix} \quad (\text{A.21})$$

where  $\mathbf{O}_{3 \times 3}$  is the null matrix, the operator  $\mathbf{S}$  is defined in equation (A.7), and  $\mathbf{M}_{ij}$  ( $i, j = 1, 2$ ) are the four  $3 \times 3$  submatrices of the global inertia matrix  $\mathbf{M}$  indexed with row and column. The  $\mathbf{C}(\boldsymbol{\nu})$  matrix is given

---

<sup>4</sup>The Coriolis effect in our case is not the one due to Earth rotation since  $\mathcal{F}_E$  is assumed to be an inertial reference frame.

explicitly in footnote<sup>5</sup>. Using this theorem (based on Kirchhoff's equations, theorem 2.2 in Fossen, 1995) to write the Coriolis matrix from the inertia one, releases us from the burden of deriving and identifying every Coriolis and centripetal effects for each of the 6 DOF. Moreover, because the inertia matrix  $M$  includes the added-mass terms,  $C(\nu)$  will also take them into account, at no additional effort. This is of utmost importance since it explains, among other effects, why an axial motion of hull shaped solid is intrinsically unstable. For instance, any small angle between the  $x$ -axis and the direction of motion will tend to increase (Munk, 1934). In such conditions, the difference between  $m'_x$  and  $m'_y$  is responsible for (not cancelling) the yaw moment induced by the Coriolis effects modelled in  $F_C$ . This resultant unstable moment is proportional to the difference ( $k_2 - k_1$ ) of the lateral and axial  $k$ -factors given in equation (A.18) and shown in figure A.6. In other words, the added-mass phenomenon not only explains why the apparent inertia of the airship is higher than that predicted by  $M_{RB}$  but is also responsible, e.g., for quite surprising behaviours due to yaw moments appearing during forward motion. Note that the Coriolis matrix also influences a number of other (maybe less significant) phenomena like slight roll angle appearing during curved trajectories.

$$5 \left( \begin{array}{cccccc} 0 & 0 & 0 & 0 & -m'_z v_z & m'_y v_y - mr_z \omega_x \\ 0 & 0 & 0 & m'_z v_z & 0 & -m'_x v_x - mr_z \omega_y \\ 0 & 0 & 0 & -m'_y v_y + mr_z \omega_x & m'_x v_x + mr_z \omega_y & 0 \\ 0 & -m'_z v_z & m'_y v_y - mr_z \omega_x & 0 & -I'_z \omega_z & mr_z v_x + I'_y \omega_y \\ m'_z v_z & 0 & -m'_x v_x - mr_z \omega_y & I'_z \omega_z & 0 & mr_z v_y - I'_x \omega_x \\ -m'_y v_y + mr_z \omega_x & m'_x v_x + mr_z \omega_y & 0 & -mr_z v_x - I'_y \omega_y & -mr_z v_y + I'_x \omega_x & 0 \end{array} \right)$$





# Appendix B

## Embedded Neural Network

### B.1 PIC-compatible Neural Network (PIC-NN)

For our lightweight robots (chapter 3), the neural network must fit the computational constraints of the embedded microcontroller (subsection 3.2.1). The *PIC-NN* (figure B.1a) is thus implemented using only integer variables with limited range, instead of using high-precision floating point values as it is usually the case when neural networks are emulated on desktop computers.

Neuron activation values (outputs) are coded as 8-bit integers in the range  $[-127,+127]$  instead of  $[-1,+1]$ , which would not be tractable without floating point data types. The PIC-NN activation function is stored in a lookup table with 255 entries (figure B.1c) so that the microcontroller does not compute the  $\tanh$  function at each update. Synapses multiply activation values by an integer factor  $w_{ij}$  in the range  $[-7,+7]$  which is then divided by 10 to ensure that a single input cannot totally saturate a neuron by itself. The range has been chosen to encode each synaptic weight in 4 bits (1 bit for the sign, 3 bits for the amplitude). Although activation values are 8-bit signed integers, the processing of the weighted sum (see figure B.1b) is done on a 16-bit signed integer to avoid overflows. The result is then limited to  $[-127,+127]$  in order to get the activation function result through the look-up table.

The PIC-NN is a discrete-time, recurrent neural network, whose computation is executed once per sensory-motor cycle. Recurrent and lateral connections take the pre-synaptic activation values from the previous cycle as input. The number of input and internal units, the number of direct connections from input to output, and the activation of lateral and recurrent connections can be freely chosen.

Each synapse of a PIC-NN is encoded on 4 bits (section B.1). The

corresponding binary genetic string is thus composed of the juxtaposition of the 4-bit blocks, each represented by a gray square in the associated connectivity matrix (figure B.1d).

## B.2 Remote and Embedded Modes

Two modes are possible with the PIC-NN. In the first one, the processing of the neural network is done in the supervising computer and the robot is only used for reading sensors and setting motor commands. We describe this mode as *remote* since the robot does not run its own controller, but is remote-controlled by the supervising computer (figure B.2a). In that case, the PIC-NN source code (written in C) is compiled within goevo. The second mode, called *embedded*, enables truly autonomous operation at the end of the evolutionary process (figure B.2b). In this mode, the supervising computer still handles the evolutionary process (subsection 6.1.2), but the neural network is embedded into the robot microcontroller.

The advantage of the remote mode is that the monitoring of the network internal state is straightforward and it is easier to debug and modify the code. However, the need for sending all sensor values at every cycle is a weakness because it takes some time (especially with vision) and lengthens the sensory-motor cycle. Furthermore, once the evolutionary process has ended the best evolved controller cannot be tested without the supervising computer, the robot is not truly autonomous. In contrast, in the embedded mode, there is a lack of visibility about internal state of the controller, but the sensory-motor cycle time can be reduced and once a genetic string is downloaded, the robot can work on its own for hours without any communication with the off-board computer.

In order to ensure the flexibility with respect to the type and the phase of experiment to be carried out, both modes are possible within our framework and can be used as required. It is also possible to do an evolutionary run in remote mode and test good controllers in embedded mode at the end. It is also very useful to have the remote mode when working with a simulated robot that does not possess a microcontroller.

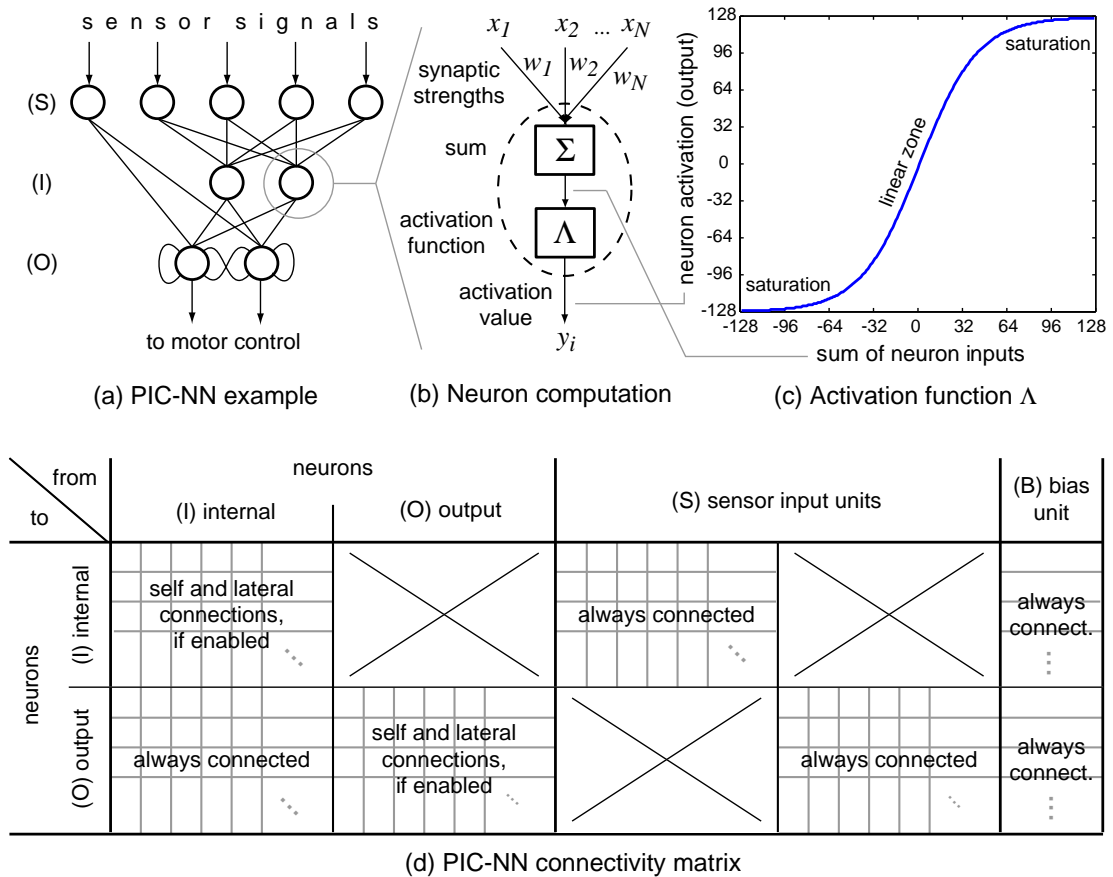


Figure B.1: The PIC-NN. (a) Architecture of the PIC-NN. Sensor input units are denoted S, input and output neurons are labelled I and O, respectively. The bias unit B is not shown. In this example, recurrent and lateral connections are present among output neurons. One input unit is directly connected to the output units, whereas four other input neurons are connected to the internal units. (b) Details of the computation occurring in a single neuron. Note that only internal and output neurons have this computation. Input units have an activation value proportional to their input. (c) Discrete activation function implemented as a lookup table in the microcontroller. (d) The PIC-NN connectivity matrix. Each gray square represents one synaptic weight. Each line corresponds either to an internal or an output neuron. Every column corresponds to one possible pre-synaptic unit: either neurons themselves, or input units, or the bias unit. The lateral and recurrent connections (on the diagonal of the left part of the matrix) can be enabled on the internal and/or output layers. In this implementation, the output neurons never send their signal back to the internal or input layers. Input units can either be connected to the internal layer or directed to the output neurons.

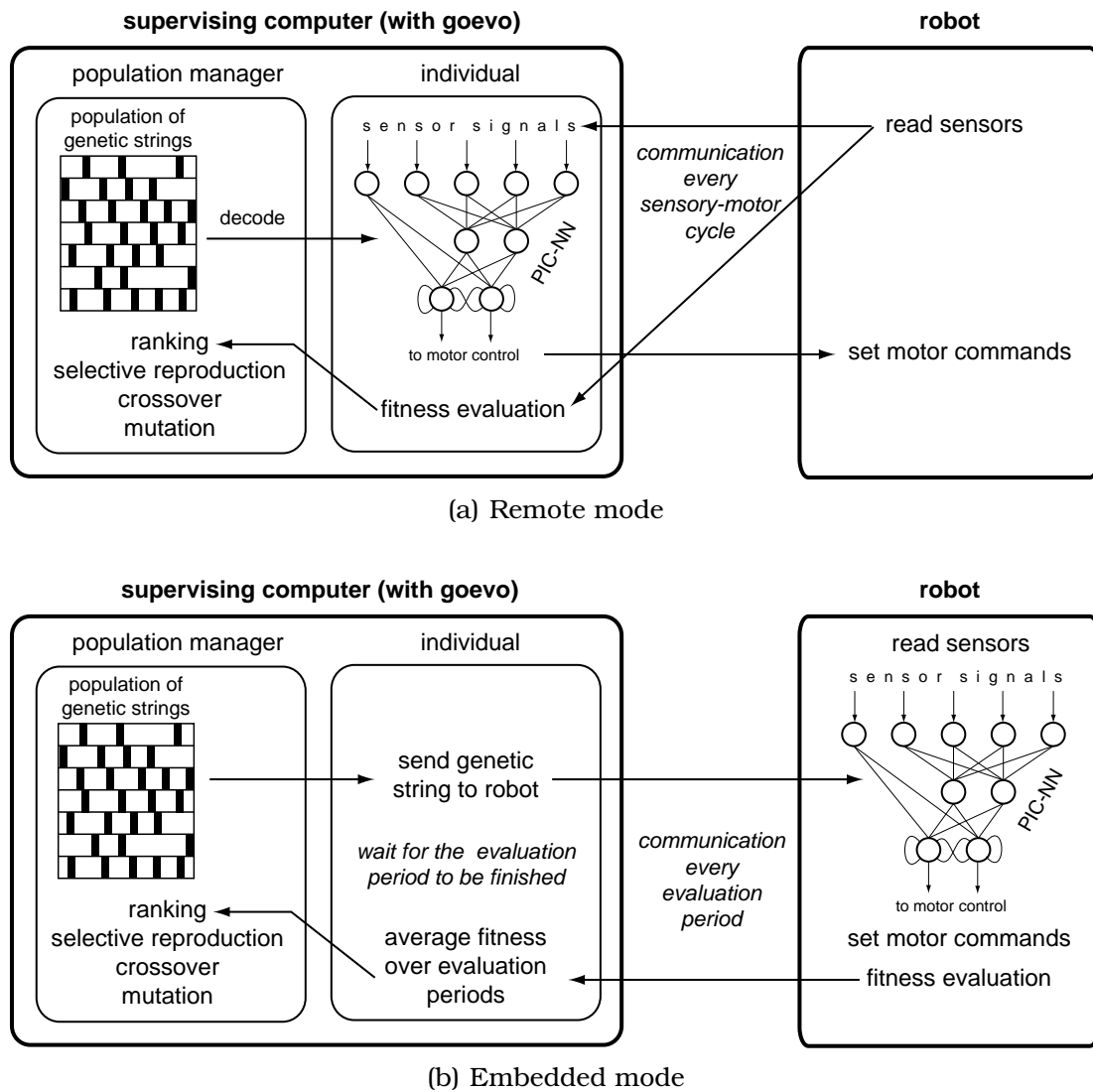


Figure B.2: Two possible configurations for evolutionary runs. (a) Remote configuration: the PIC-NN is run in the supervising computer that asks the robot for sensor values at the beginning of every sensory-motor cycle and sends back the motor commands to the robot. (b) Embedded mode: PIC-NN is embedded in the robot microcontroller and communication occurs only at the beginning of an evaluation and at the end.

# Bibliography

- N. Ancona and T. Poggio. Optical flow from 1d correlation: Application to a simple time-to-crash detector. In *Proceedings of Fourth International Conference on Computer Vision*, pages 209–214, 1993.
- S. Avadhanula, R.J. Wood, E. Steltz, J. Yan, and R.S. Fearing. Lift force improvements for the micromechanical flying insect. In *Proceeding of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.
- A. Bab-Hadiashar, D. Suter, and R. Jarvis. Image-interpolation based optic flow technique. Technical Report MECSE-96-1, Monash University, Clayton, Australia, 1996.
- J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- G.L. Barrows, J.S. Chahl, and M.V. Srinivasan. Biomimetic visual sensing and flight control. In *Bristol Conference on UAV Systems*, 2002.
- G.L. Barrows, C. Neely, and K.T. Miller. Optic flow sensors for mav navigation. In Thomas J. Mueller, editor, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, volume 195 of *Progress in Astronautics and Aeronautics*, pages 557–574. AIAA, 2001.
- A. Borst. How do flies land? from behavior to neuronal circuits. *BioScience*, 40(4):292–299, 1990.
- A. Borst and S. Bahde. Visual information processing in the fly’s landing system. *Journal of Comparative Physiology A*, 163:167–173, 1988.
- A. Borst and M. Egelhaaf. Principles of visual motion detection. *Trends in Neurosciences*, 12(8):297–306, 1989.
- A. Borst, M. Egelhaaf, and H. S. Seung. Two-dimensional motion perception in flies. *Neural Computation*, 5(6):856–868, 1993.

- S. Bouabdallah, P. Murrieri, and R. Siegwart. Towards autonomous indoor micro vtol. *Autonomous Robots*, 2005. In press.
- V. Braitenberg. *Vehicules - experiments in synthetic psychology*. The MIT Press, Cambridge, MA, 1984.
- R.A. Brooks. Achieving artificial intelligence through building robots. Technical Report A.I. Memo 899, Massachusetts Institute of Technology, Cambridge, MA, 1986.
- R.A. Brooks. *Cambrian Intelligence*. The MIT Press, Cambridge, MA, 1999.
- C. Cafforio and F. Rocca. Methods for measuring small displacements of television images. *IEEE Transactions on Information Theory*, 22: 573– 579, 1976.
- T. Camus. Calculating time-to-contact using real-time quantized optical flow. Technical Report 5609, National Institute Of Standards and Technology NISTIR, 1995.
- J.S. Chahl and M.V. Srinivasan. Visual computation of egomotion using an image interpolation technique. *Biological Cybernetics*, 74:405–411, 1996.
- J.S. Chahl, M.V. Srinivasan, and H. Zhang. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *The International Journal of Robotics Research*, 23(2):101–1102, 2004.
- W.P. Chan, F. Prete, and M.H. Dickinson. Visual input to the efferent control system of a fly's "gyroscope". *Science*, 280(5361):289–292, 1998.
- C. Chang and P. Gaudiano, editors. *Biomimetic Robotics*, volume 30 of *Robotics and Autonomous Systems, Special Issue*. Elsevier, 2000.
- R.F. Chapman. *The Insects: Structure and Function*. Cambridge University Press, 4th edition, 1998.
- D. Cliff, I. Harvey, and P. Husbands. Artificial evolution of visual control systems for robots. In M. Srinivisan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines*, pages 126–157. Oxford University Press, 1997.

- D. Cliff and G.F. Miller. Co-evolution of pursuit and evasion ii: Simulation methods and results. In P. Maes, M. Mataric, J.A. Meyer, J. Pollack, H. Roitblat, and S. Wilson, editors, *From Animals to Animats IV: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press-Bradford Books, 1996.
- T.S. Collett and M.F. Land. Visual control of flight behavior in the hoverfly, *syricta pipiens*. *Journal of Comparative Physiology*, 99:1–66, 1975.
- D. Coombs, M. Herman, T.H. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. In *Proceedings of the Fifth International Conference on Computer Vision*, 1995.
- F.M. da Silva Metelo and L.R. Garcia Campos. Vision based control of an autonomous blimp. Technical report, 2003.
- C.T. David. Compensation for height in the control of groundspeed by *drosophila* in a new, 'barber's pole' wind tunnel. *Journal of Comparative Physiology A*, 147:485–493, 1982.
- A.F.P. de Talens and C.T. Ferretti. Landing and optomotor responses of the fly *musca*. In G.A. Horridge, editor, *The Compound Eye and Vision of Insects*, pages 490–501. Clarendon Press, Oxford, 1975.
- T. Delbrück and C.A. Mead. Analog vlsi phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits. In C. Koch and H. Li, editors, *Vision Chips: Implementing vision algorithms with analog VLSI circuits*, pages 139–161. IEEE Computer Society Press, 1995.
- M.H. Dickinson. Haltere-mediated equilibrium reflexes of the fruit fly, *drosophila melanogaster*. *Philosophical Transactions: Biological Sciences*, 354(1385):903–916, 1999.
- M.H. Dickinson, F.O. Lehmann, and S.P. Sane. Wing rotation and the aerodynamic basis of insect flight. *Science*, 284:1954–1960, 1999.
- J.K. Douglass and N.J. Strausfeld. Visual motion-detection circuits in flies: parallel direction- and non-direction-sensitive pathways between the medulla and lobula plate. *Journal of Neuroscience*, 16(15):4551–4562, 1996.

- R.O. Dror, D.C. O'Carroll, and S.B. Laughlin. Accuracy of velocity estimation by reichardt correlators. *Journal of Optical Society of America A*, 18:241–252, 2001.
- A.P. Duchon and W. H. Warren. Robot navigation from a gibsonian viewpoint. In *Proceedings of IEEE Conference on Systems, Man and Cybernetics, San Antonio (TX)*, pages 2272–2277, 1994.
- A.P. Duchon, W. H. Warren, and L.P. Kaelbling. Ecological robotics. *Adaptive Behavior*, 6:473–507, 1998.
- R. Dudley. *The Biomechanics of Insect Flight: Form, Function, Evolution*. Princeton University Press, 2000.
- M. Egelhaaf and A. Borst. Transient and steady-state response properties of movement detectors. *Journal of Optical Society of America A*, 6(1):116–127, 1989.
- M. Egelhaaf and A. Borst. A look into the cockpit of the fly: Visual orientation, algorithms, and identified neurons. *The Journal of Neuroscience*, 13(11):4563–4574, 1993a.
- M. Egelhaaf and A. Borst. Motion computation and visual orientation in flies. *Comparative biochemistry and physiology*, 104A(4):659–673, 1993b.
- M. Egelhaaf and R. Kern. Vision in flying insects. *Current Opinion in Neurobiology*, 12(6):699–706, 2002.
- M. Egelhaaf, R. Kern, H.G. Krapp, J. Kretzberg, R. Kurtz, and A.K. Warzechna. Neural encoding of behaviourally relevant visual-motion information in the fly. *TRENDS in Neurosciences*, 25(2):96–102, 2002.
- S.M. Ettinger, M.C. Nechyba, P.G. Ifju, and M. Waszak. Vision-guided flight stability and control for micro air vehicles. *Advanced Robotics*, 17(3):617–40, 2003.
- R.S. Fearing, K.H. Chiang, M.H. Dickinson, D.L. Pick, M. Sitti, and J. Yan. Wing transmission for a micromechanical flying insect. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2000.
- C. Fennema and W.B. Thompson. Velocity determination in scenes containing several moving objects. *Computer graphics and image processing*, 9:301–315, 1979.



- C. Fermüller and Y. Aloimonos. Primates, bees, and uavs (unmanned ground vehicles) in motion. In M. Srinivisan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines*, pages 199–225. Oxford University Press, 1997.
- D. Floreano, T. Kato, D. Marocco, and E. Sauser. Coevolution of active vision and feature selection. *Biological Cybernetics*, 90(3):218–228., 2004.
- D. Floreano and C. Mattiussi. Evolution of spiking neural controllers for autonomous vision-based robots. In T. Gomi, editor, *Evolutionary Robotics IV*, pages 38–61. Springer-Verlag, 2001.
- D. Floreano, N. Schoeni, G. Caprari, and J. Blynel. Evolutionary bits'n'spikes. In R. K. Standish, M. A. Beadau, and H. A. Abbass, editors, *Artificial Life VIII: Proceedings of the Eight International Conference on Artificial Life*. MIT Press, 2002.
- D. Floreano, J.C. Zufferey, and C. Mattiussi. Evolving spiking neurons from wheels to wings. In K. Murase and T. Asakura, editors, *Dynamic Systems Approach for Embodiment and Sociality*, volume 6 of *Advanced Knowledge International, International Series on Advanced Intelligence*, pages 65–70, 2003.
- D. Floreano, J.C. Zufferey, and J.D. Nicoud. From wheels to wings with evolutionary spiking neurons. *Artificial Life*, 11(1-2), 2005. In press.
- T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, New York, 1995.
- N. Franceschini. Sampling of the visual environment by the compound eye of the fly: Fundamentals and applications. In A. W. Snyder and R. Menzel, editors, *Photoreceptor Optics*, pages 98–125. Springer, Berlin, 1975.
- N. Franceschini. From fly vision to robot vision: Re-construction as a mode of discovery. In Secomb T.W. Barth F.G., Humphrey J.A., editor, *Sensors and Sensing in Biology and Engineering*, pages 223–235. Springer, 2003.
- N. Franceschini. Visual guidance based on optic flow: a biorobotic approach. *Journal of Physiology - Paris*, 98:281–292, 2004.

- N. Franceschini, J.M. Pichon, and C. Blanes. From insect vision to robot vision. *Philosophical Transactions of the Royal Society B*, 337: 283–294, 1992.
- N. Franceschini, A. Riehle, and A. Le Nestour. Directionally selective motion detection by insect neurons. In D.G. Stavenga and R.C. Hardie, editors, *Facets of Vision*, pages 360–390. Springer-Verlag, 1989.
- M.O. Franz and J.S. Chahl. Insect-inspired estimation of self-motion. In *Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision, LNCS*. Springer-Verlag, 2002.
- M.O. Franz and H.G. Krapp. Wide-field, motion-sensitive neurons and matched filters for optic flow fields. *Biological Cybernetics*, 83:185–197, 2000.
- M.O. Franz and H.A. Mallot. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30:133–153, 2000.
- J.J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, 1950.
- J.J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979.
- D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- J.M. Grasmeyer and M.T. Keennon. Development of the black widow micro air vehicle. In Thomas J. Mueller, editor, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, volume 195 of *Progress in Astronautics and Aeronautics*, pages 519–535. AIAA, 2001.
- W.E. Green, P.Y. Oh, and G.L. Barrows. Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2004.
- K.G. Götz. The optomotor equilibrium of the drosophila navigation system. *Journal of Comparative Physiology*, 99:187–210, 1975.
- A. Guanella. Flight simulator for robotic indoor slow flyers. Diploma project, EPFL, 2004.

- J. Haag, M. Egelhaaf, and A. Borst. Dendritic integration of motion information in visual interneurons of the blowfly. *Neuroscience Letters*, 140:173–176, 1992.
- R.R. Harrison. *An Analog VLSI Motion Sensor Based on the Fly Visual System*. PhD thesis, 2000.
- R.R. Harrison. An algorithm for visual collision detection in real-world scenes. Technical report, 2003.
- R.R. Harrison and C. Koch. A robust analog vlsi motion sensor based on the visual system of the fly. *Autonomous Robots*, 7:211–224, 1999.
- I. Harvey, P. Husbands, and D. Cliff. Seeing the light: Artificial evolution, real vision. In D. Cliff, P. Husbands, J. Meyer, and S. Wilson, editors, *From Animals to Animats III*, pages 392–401. The MIT Press, 1994.
- B. Hassenstein and W. Reichardt. Systemtheoretische analyse der zeit-, reihenfolgen- und vorzeichenbewertung bei der bewegungsperzeption des rüsselkäfers chlorophanus. *Zeitschrift für Naturforschung*, 11b:513–524, 1956.
- K. Hausen. Motion sensitive interneurons in the optomotor system of the fly. ii. the horizontal cells: Receptive field organization and response characteristics. *Biological Cybernetics*, 46:67–79, 1982.
- K. Hausen and M. Egelhaaf. Neural mechanisms of visual course control in insects. In D.G. Stavenga and R.C. Hardie, editors, *Facets of Vision*. Springer-Verlag, 1989.
- R. Hengstenberg. Gaze control in the blowfly calliphora: A multisensory two-stage integration process. *The Neuroscience*, 3:19–29, 1991.
- B.K. Horn. *Robot vision*. MIT Press, 1986.
- B.K. Horn and P. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- A. Horridge. Insects which turn and look. *Endeavour*, 1:7–17, 1977.
- A. Horridge. Pattern and 3d vision of insects. In Y. Aloimonos, editor, *Visual Navigation*, pages 26–59. Lawrence Erlbaum Associates, Mahwah, New Jersey, 1997.

- S.A. Huber. *Studies of the Visual Orientation Behavior in Flies Using the Artificial Life Approach*. PhD thesis, Eberhard-Karls-Universität zu Tübingen, 1997.
- S.A. Huber, H.A. Mallot, and H.H. Bülthoff. Evolution of the sensorimotor control in an autonomous agent. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 449–457. MIT Press, 1996.
- F. Iida. Goal-directed navigation of an autonomous flying robot using biologically inspired cheap vision. In *Proceedings of the 32nd International Symposium on Robotics*, 2001.
- F. Iida. Biologically inspired visual odometer for navigation of a flying robot. *Robotics and Autonomous Systems*, 44:201–208, 2003.
- F. Iida and D. Lambrinos. Navigation in an autonomous flying robot by using a biologically inspired visual odometer. In *Sensor Fusion and Decentralized Control in Robotic System III, Photonics East, Proceeding of SPIE*, volume 4196, pages 86–97, 2000.
- G. Indiveri, J. Kramer, and C. Koch. System implementations of analog vlsi velocity sensors. *IEEE Micro*, 16:40–49, 1996.
- K.D. Jones, C.J. Bradshaw, J. Papadopoulos, and M.F. Platzer. Improved performance and control of flapping-wing propelled micro air vehicles. In *Proceedings of the 42nd Aerospace Sciences Meeting and Exhibit, Reno NV*, 2004.
- K. Karmeier, M. Egelhaaf, and H.G. Krapp. Early visual experience and receptive field organization of the optic flow processing interneurons in the fly motion pathway. *Visual Neuroscience*, 18:1–8, 2001.
- J.S. Kennedy. The migration of the desert locust. *Philosophical transactions of the Royal Society of London B*, 235:163–290, 1951.
- G.A. Khoury and J.D. Gillet. *Airship Technology*. Cambridge University Press, London, 1999.
- J.J. Koenderink and A.J. van Doorn. Facts on optic flow. *Biological Cybernetics*, 56:247–254, 1987.
- J. Kramer, R. Sarpeshkar, and C. Koch. An analog vlsi velocity sensor. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 413–416, 1995.

- H.G. Krapp. Neuronal matched filters for optic flow processing in flying insects. In M. Lappe, editor, *Neuronal Processing of Optic Flow*, pages 93–120. San Diego: Academic Press, 2000.
- H.G. Krapp, B. Hengstenberg, and R. Hengstenberg. Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *Journal of Neurophysiology*, 79:1902–1917, 1998.
- H.G. Krapp and R. Hengstenberg. Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, 384:463–466, 1996.
- I. Kroo and P. Kunz. Mesoscale flight and miniature rotorcraft development. In Thomas J. Mueller, editor, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, volume 195 of *Progress in Astronautics and Aeronautics*, pages 503–517. AIAA, 2001.
- H. Lamb. *Hydrodynamics*. Cambridge University Press, London, 1932.
- D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems*, 30:39–64, 2000.
- M.F. Land. Visual acuity in insects. *Annual Review of Entomology*, 42:147–177, 1997.
- D.N. Lee. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5:437–459, 1976.
- D.N. Lee and P.E. Reddish. Plummeting gannets: a paradigm of ecological optics. *Nature*, 293:193–294, 1981.
- F.O. Lehmann. Aerial locomotion in flies and robots: kinematic control and aerodynamics of oscillating wings. *Arthropod Structure & Development*, 33:331–345, 2004.
- M.A. Lewis. Visual navigation in a robot using zig-zag behavior. In *Neural Information Processing Systems 10*. MIT Press, 1998.
- L. Lichtensteiger and P. Eggenberger. Evolving the morphology of a compound eye on a robot. In *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot '99)*, pages 127–134, 1999.

- J.P. Lindemann, R. Kern, C. Michaelis, J.-A. Meyer, J.H. van Hateren, and M. Egelhaaf. Flimax, a novel stimulus device for panoramic and highspeed presentation of behaviourally generated optic flow. *Vision Research*, 43:779–791, 2003.
- S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, and R. Douglas. *Analog VLSI: Circuits and Principles*. MIT Press, Cambridge, MA, 2003.
- B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver*, pages 674–679, 1981.
- H.A. Mallot. *Computational Vision: Information Processing in Perception and Visual Behavior*. The MIT Press, 2000.
- D. Marocco and D. Floreano. Active vision and feature selection in evolutionary behavioral systems. In J. Hallam, D. Floreano, G. Hayes, and J. Meyer, editors, *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*, pages 247–255, Cambridge, MA, 2002. MIT Press-Bradford Books.
- D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, New York, 1982.
- J.H. McMasters and M.L. Henderson. Low speed single element airfoil synthesis. *Technical Soaring*, 2:1–21, 1980.
- C. Melhuish and J. Welsby. Gradient ascent with a group of minimalist real robots: Implementing secondary swarming. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002.
- O. Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: a tool for investigation in control algorithms. In T. Yoshikawa and F. Miyazaki, editors, *Proceedings of the Third International Symposium on Experimental Robotics, Kyoto, Japan*, pages 501–513. Springer Verlag, 1993.

- T.J. Mueller. *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, volume 195 of *Progress in Astronautics and Aeronautics*. AIAA, 2001.
- T.J. Mueller and J.D. DeLaurier. An overview of micro air vehicle. In T.J. Mueller, editor, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, *Progress in Astronautics and Aeronautics*, pages 1–10. AIAA, 2001.
- M.M. Munk. Fluid mechanics, second part. In W.F. Durand, editor, *Aerodynamic Theory I*, pages 224–304. Julius Springer, 1934.
- M.M. Munk. Aerodynamics of airships. In W.F. Durand, editor, *Aerodynamic Theory VI*, pages 32–48. Julius Springer, 1936.
- F. Mura and N. Franceschini. Visual control of altitude and speed in a flying agent. In *From Animals to Animats III*, pages 91–99. MIT Press, 1994.
- L. Muratet, S. Doncieux, Y. Brière, and J.-A. Meyer. A contribution to vision-based autonomous helicopter flight in urban environments. *To appear in Robotics and Autonomous Systems*, 2005.
- H.H. Nagel. On change detection and displacement vector estimation in image sequences. *Pattern Recognition Letters*, 1:55–59, 1982.
- M.G. Nagle and M.V. Srinivasan. Structure from motion: determining the range and orientation of surfaces by image interpolation. *Journal of the Optical Society of America A*, 13(1):5–34, 1996.
- G. Nalbach. The halteres of the blowfly calliphora i. kinematics and dynamics. *Journal of Comparative Physiology A.*, 1993.
- G. Nalbach. Extremely non-orthogonal axes in a sense organ for rotation: Behavioural analysis of the dipteran haltere system. *Neuroscience*, 61:49–163, 1994.
- G. Nalbach and R. Hengstenberg. The halteres of the blowfly calliphora - three-dimensional organization of compensatory reactions to real and simulated rotations. *Journal of Comparative Physiology A*, 175: 695–708, 1994.
- R.C. Nelson and Y. Aloimonos. Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.

- R.C. Nelson and Y. Aloimonos. Obstacle avoidance using flow field divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1102–1106, 1989.
- T. Netter and N. Franceschini. A robotic aircraft that follows terrain using a neuromorphic eye. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- T.R. Neumann. Modeling insect compound eyes: Space-variant spherical vision. In T. Poggio H. H. Bülthoff, S.-W. Lee and C. Wallraven, editors, *Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision*. Springer-Verlag, 2002.
- T.R. Neumann and H.H. Bülthoff. Insect inspired visual control of translatory flight. In J. Kelemen and P. Sosik, editors, *Advances in Artificial Life : 6th European Conference, ECAL 2001, Prague, Czech Republic*, volume 2159 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001.
- T.R. Neumann and H.H. Bülthoff. Behavior-oriented vision for biomimetic flight control. In *Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics*, pages 196–203, 2002.
- T.R. Neumann, S.A. Huber, and H.H. Bülthoff. Minimalistic approach to 3d obstacle avoidance behavior from simulated evolution. In *Proceedings of the 7th International Conference on Artificial Neural Networks (ICANN)*, volume 1327 of *Lecture Notes in Computer Science*, pages 715–720. Springer-Verlag, 1997.
- J.D. Nicoud and J.C. Zufferey. Toward indoor flying robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 787–792, 2002.
- S. Nolfi and D. Floreano. *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-organizing Machines*. MIT Press, Cambridge, MA, 2000.
- R. Pfeifer and D. Lambrinos. Cheap vision - exploiting ecological niche and morphology. In V. Hlava, K.G. Jeffery, and J. Wiedermann, editors, *SOFSEM 2000: Theory and Practice of Informatics: 27th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic.*, volume 1963 of *Lecture Notes in Computer Science*, pages 202–226, 2000.



- J.M. Pichon, C. Blanes, and N. Franceschini. Visual guidance of a mobile robot equipped with a network of self-motion sensors. *Proceedings of SPIE: Mobile Robots IV*, 1195:44–53, 1990.
- C. Planta, J. Conradt, A. Jencik, and P. Verschure. A neural model of the fly visual system applied to navigational tasks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2002.
- T. Poggio, A. Verri, and V. Torre. Green theorems and qualitative properties of the optical flow. Technical Report A.I. Memo 1289, Massachusetts Institute of Technology, Cambridge, MA, 1991.
- T.N. Pornsin-Sirirak, Y.-Ch. Tai, and Ch.-M. Ho. Microbat: A palm-sized electrically powered ornithopter. In *Proceedings of NASA/JPL Workshop on Biomimetic Robotics*, 2001.
- W. Reichardt. Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In W. A. Rosenblith, editor, *Sensory Communication*, pages 303–317. Wiley, New York, 1961.
- W. Reichardt. Movement perception in insects. In W. Reichardt, editor, *Processing of optical data by organisms and by machines*, pages 465–493. New York: Academic Press, 1969.
- M.B. Reiser and M.H. Dickinson. A test bed for insect-inspired robotic control. *Philosophical Transactions: Mathematical, Physical & Engineering Sciences*, 361:2267–2285, 2003.
- F.C. Rind. Motion detectors in the locust visual system: From biology to robot sensors. *Microscopy Research and Technique*, 56(4):256 – 269, 2002.
- F. Ruffier. *Pilote automatique biomimetique*. PhD thesis, 2004.
- F. Ruffier and N. Franceschini. Octave, a bioinspired visuo-moto control system for the guidance of micro-air-vehicles. In A. Rodriguez-Vazquez, D. Abott, and R. Carmona, editors, *Proceedings of SPIE Conference on Bioengineered and Bioinspired Systems*, pages 1–12, 2003.
- F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans*, pages 2339–2346. T. J. Tarn and T. Fukuda and K. Valavanis, April 2004.

- S.I. Sagatun and T. Fossen. Lagrangian formulation of underwater vehicles' dynamics. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1029–1034, 1991.
- J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: A step forward to a robotic bee. *International Journal of Computer Vision*, 14:159–177, 1995.
- L. Schenato, W.C. Wu, and S.S. Sastry. Attitude control for a micro-mechanical flying insect via sensor output feedback. *IEEE Journal of Robotics and Automation*, 20(1):93–106, 2004.
- C. Schilstra and J.H. van Hateren. Blowfly flight and optic flow. i. thorax kinematics and flight dynamics. *Journal of Experimental Biology*, 202:1481–1490, 1999.
- H. Schuppe and R. Hengstenberg. Optical properties of the ocelli of calliphora erythrocephala and their role in the dorsal light response. *Journal of Comparative Physiology A*, 173:143–149, 1993.
- A. Sherman and M.H. Dickinson. Summation of visual and mechanosensory feedback in drosophila flight control. *Journal of Experimental Biology*, 207:133–142, 2004.
- R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robotics*. MIT Press, 2004.
- S. Single, J. Haag, and A. Borst. Dendritic computation of direction selectivity and gain control in visual interneurons. *Journal of Neuroscience*, 17:6023–6030, 1997.
- E. Sobel. The locust's use of motion parallax to measure distance. *Journal of Comparative Physiology A*, 167:579–588, 1990.
- P. Sobey. Active navigation with a monocular robot. *Biological Cybernetics*, 71:433–440, 1994.
- M.V. Srinivasan. An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71:401–416, 1994.
- M.V. Srinivasan. Visual navigation: The eyes know where their owner is going. In *Motion vision: computational, neural and ecological constraints*, pages 180–186. Springer, 2000.

- M.V. Srinivasan, J.S. Chahl, M.G. Nagle, and S.W. Zhang. Embodying natural vision into machines. In M.V. Srinivasan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines*, pages 249–265. 1997.
- M.V. Srinivasan, J.S. Chahl, K. Weber, S. Venkatesh, and H. Zhang. Robot navigation inspired by principles of insect vision. In A. Zelinsky, editor, *Field and Service Robotics*, pages 12–16. Springer-Verlag, 1998.
- M.V. Srinivasan, M. Lehrer, W.H. Kirchner, and S.W. Zhang. Range perception through apparent image speed in freely-flying honeybees. *Visual Neuroscience*, 6:519–535, 1991.
- M.V. Srinivasan, M. Poteser, and K. Kral. Motion detection in insect orientation and navigation. *Vision Research*, 39(16):2749–2766, 1999.
- M.V. Srinivasan and H. Zhang. Visual navigation in flying insects. *International review of neurobiology*, 44:67–92, 2000.
- M.V. Srinivasan, S.W. Zhang, J.S. Chahl, E. Barth, and S. Venkatesh. How honeybees make grazing landings on flat surfaces. *Biological Cybernetics*, 83:171–183, 2000.
- M.V. Srinivasan, S.W. Zhang, M. Lehrer, and T.S. Collett. Honeybee navigation en route to the goal: Visual flight control and odometry. *The Journal of Experimental Biology*, 199:237–244, 1996.
- L.F. Tammero and M.H. Dickinson. The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *The Journal of Experimental Biology*, 205:327–343, 2001.
- L.F. Tammero and M.H. Dickinson. Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly. *The Journal of Experimental Biology*, 205:2785–2798, 2002a.
- L.F. Tammero and M.H. Dickinson. The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *The Journal of Experimental Biology*, 205:327–343, 2002b.
- J. Urzelai and D. Floreano. Evolution of adaptive synapses: robots with fast adaptive behavior in new environments. *Evolutionary Computation*, 9:495–524, 2001.

- S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39:145–155, 2002.
- J.H. van Hateren and C. Schilstra. Blowfly flight and optic flow. ii. head movements during flight. *Journal of Experimental Biology*, 202: 1491–1500, 1999.
- A. Verri, M. Straforini, and V. Torre. Computational aspects of motion perception in natural and artificial vision systems. *Philosophical Transactions of the Royal Society B*, 337:429–443, 1992.
- H. Wagner. Flow-field variables trigger landing in flies. *Nature*, 297: 147–148, 1982.
- H. Wagner. Flight performance and visual control of flight of the free-flying housefly (*Musca domestica* L.). i. organization of the flight motor. *Philosophical Transactions of the Royal Society B*, 312:527–551, 1986.
- B. Webb and T.R. Consi. *Biorobotics: Methods and Applications*. MIT Press, 2001.
- B. Webb, R.R. Harrison, and M.A. Willis. Sensorimotor control of navigation in arthropod and artificial systems. *Arthropod Structure & Development*, 33:301–329, 2004.
- K. Weber, S. Venkatesh, and M.V. Srinivasan. Insect inspired behaviours for the autonomous control of mobile robots. In M. V. Srinivasan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines*, pages 226–248. Oxford University Press, 1997.
- R. Wehner. Matched filters - neural models of the external world. *Journal of Comparative Physiology A*, 161:511–531, 1987.
- T.C. Whiteside and G.D. Samuel. Blur zone. *Nature*, 225:94–5, 1970.
- W.C. Wu, L. Schenato, R.J. Wood, and R.S. Fearing. Biomimetic sensor suite for flight control of a micromechanical flight insect: Design and experimental results. In *Proceeding of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, 2003.
- H. Zhang and J. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. Technical report, 1998.

- J.C. Zufferey, A. Beyeler, and D. Floreano. Vision-based navigation from wheels to wings. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- J.C. Zufferey and D. Floreano. Optic-flow-based steering and altitude control for ultra-light indoor aircraft. Technical report, 2004.
- J.C. Zufferey and D. Floreano. Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. In *Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona*, 2005.
- J.C. Zufferey and D. Floreano. Fly-inspired visual steering of an ultra-light indoor aircraft. *IEEE Transactions on Robotics*, To appear.
- J.C. Zufferey, D. Floreano, M. van Leeuwen, and T. Merenda. Evolving vision-based flying robots. In Bülthoff, Lee, Poggio, and Wallraven, editors, *Biologically Motivated Computer Vision: Second International Workshop, BMCV 2002, Tübingen, Germany*, volume 2525 of *Lecture Notes in Computer Science*, pages 592–600. Springer-Verlag, 2002.
- J.C. Zufferey, C. Halter, and J.D. Nicoud. Avion d'intérieur : une plateforme de recherche pour la robotique bio-inspirée. *Modell Flugsport (Swiss Magazine)*, December 2001.

# Index

- ADC, *see* Analog to Digital Converter
- Aerodynamic damping, 150
- ALC, *see* Altitude Control
- Altitude Control, 42, 101
- Analog to Digital Converter, 58
- Angular velocity, 76
- Aperture problem, 72
- Apis, 21
- Archimedes' principle, 149
- Arthropods, 19
- Artificial evolution
  - crossover, 120
  - elitism, 120
  - evaluation period, 120
  - fitness function, 122
  - genetic string, 120
  - individual, 120
  - mutation, 120
  - population, 120
- ATC, *see* Attitude Control
- Attitude, 39
- Attitude Control, 39
- aVLSI, 146
- Azimuth (angle), 73
- Blimp2b: the name of the indoor air-ship, 49
- Bluetooth, 64
- Body-fixed frame, 147
- Calliphora, 20
- Camera, 60
- CMOS, 60
- Compound eye, 23
- Contrast Rate Detector, 129
- Coriolis forces, 158
- Course Stabilisation, 40, 98
- CRD, *see* Contrast Rate Detector
- CS, *see* Course Stabilisation
- Degrees of Freedom, 47, 147
- Dihedral, 40, 56
- Diptera, 20
- DOF, *see* Degrees of Freedom
- Drosophila, 21
- Elementary Motion Detector, 30
  - dependency on contrast, 31, 78
  - dependency on spatial frequency, 31, 78
  - spatial summation, 78
- Elevation (angle), 73
- Elevator, 55, 107
- EMD, *see* Elementary Motion Detector
- ER, *see* Evolutionary Robotics
- Evolutionary Robotics, 13, 118
- F2: the name of the 30-grame indoor airplane, 54
- FOC, *see* Focus of contraction
- Focus
  - of contraction, 75
  - of expansion, 75, 90
- FOE, *see* Focus of expansion
- GAs, *see* Genetic Algorithms
- Genetic Algorithms, 13
- Gyroscope, 63
- Halteres, 21, 25
- HCI, *see* Bluetooth

- I2A, *see* Image Interpolation Algorithm
- Image Interpolation Algorithm, 79
- Inertia of an airship, 152
- Inertial frame, 147
- Khepera: the name of the miniature wheeled robot, 46
- Lamina, 28
- Linear camera, *see* Camera
- Lobula (plate, complex), 29
- Lobula complex, 28
- LPTC, *see* Tangential cells
- Matched filters, 35, 78
- MAV, *see* Micro Air Vehicle
- Medulla, 28, 29
- MEMS, *see* Gyroscope
- Mercator map, projection, 74
- Micro Air Vehicle, 7, 52, 145
- Microcontroller, 58
- Motion field, 72
- Motion parallax, 76
- Musca, 20
- Neural network, 121
  - activation function, 121
  - activation value, 121
  - artificial neuron, 121
  - bias unit, 121
  - synaptic strength, 121
- OA, *see* Obstacle Avoidance
- Obstacle Avoidance, 41, 98
- Ocelli, 25
- OF, *see* Optic Flow
- OFD, *see* Optic Flow Detector
- Ommatidia, 20
- Optic Flow, 71
  - characteristic patterns, 89
  - constraint equation, 79
  - detection, 77
  - formal description, 73
  - gradient-based methods, 78
  - image interpolation algorithm, 81
  - local constancy assumption, 79
  - smoothness constraint, 79
- Optic Flow Detector, 82
- PIC, *see* Microcontroller
- PIC-NN (PIC compatible neural network), 161
- Reference image, 81
- Restoring forces, 149
- Rotatory
  - cancellation of RotOF, 103
  - component of optic flow, 74
- RotOF, *see* Rotatory component of optic flow
- Rudder, 55, 107
- Saccade, 42, 99
- Spatio-Temporal Integration of Motion, 38, 99
- Steering Control, 98
- STIM, *see* Spatio-Temporal Integration of Motion
- Tangential cells, 29
- Time To Contact, 37, 92
- Translatory
  - component of optic flow, 74
  - motion, 76, 90
- TransOF, *see* Translatory component of optic flow
- TTC, *see* Time To Contact
- UAV, *see* Unmanned Aerial Vehicle
- Unmanned Aerial Vehicle, 3, 145
- Very Large Scale Integration, 31, 146
- Virtual sensors, 66
- VLSI, *see* Very Large Scale Integration





# Curriculum Vitae

Born on 18 October 1974, I grew up in Chippis, Valais, Switzerland. In 1995, I graduated from the Collège des Creusets in Sion, high school in science (type C). The same year, I entered the Swiss Federal Institute of Technology in Lausanne (EPFL). After the first year, I took a sabbatical devoted to the Swiss military pilot school in Magadino, Ticino.

Back at EPFL in 1997, I focused my studies on behaviour-based and miniature mobile robotics. After completing an exchange semester at the Carnegie Mellon University in Pittsburg (CMU), where I worked on a hopping robot for my diploma, I graduated in 2001 with a M.S. in micro-engineering. Since May 2001, I pursue my Ph.D. on Bio-inspired Vision-based Flying Robots as a research assistant at the Autonomous Systems Laboratory (EPFL), under the supervision of Prof. Dario Floreano. In parallel to my research activity, I am involved in educational tasks such as the preparation of several practical assignments with robots for undergraduate students.

My main fields of interests are aviation and autonomous robotics. In 1993, I obtained the glider licence, followed by the single-engine private licence in 1997, and aerobatic extension in 2003. In 2001, I co-founded an EPFL spin-off company, DIDEL SA, which is involved in educational robotics and ultra-light indoor slow-flyers. I initiated the flying robot contest at EPFL and co-organised it in 2004 and 2005.

Dans la certitude de quels ciels,  
au coeur de quels frêles espaces,  
êtreindre les chauds reflets de vivre ?

Frémis,  
Matière qui t'éveilles,  
scintille plutôt que ne luis,  
tremble comme le milieu de la flamme,  
Matière qui virevoltes et t'enfuis  
et,  
parmi les vents illimités de la conscience,  
aspire à être...

*Julien*