



INSTITUTO SUPERIOR TÉCNICO  
Universidade Técnica de Lisboa

# **Sensor Based Control of a Quadrotor**

**Pedro João Monteiro Grilo Lema Santos**

A thesis submitted to the Department of Electrotechnical Engineering in  
partial fulfilment of the requirements for the degree of

## **Master of Science in Electrical and Computer Engineering**

### **Panel**

President: Prof. Carlos Bispo

Advisor: Prof. Carlos Silvestre

Co-Advisor: Prof. Paulo Oliveira

Examiner: Prof. Rodrigo Ventura

**October 2011**



# Agradecimentos

A minha dissertação de mestrado, a qual apresento neste documento, prolongou-se-se por cerca de 7 meses. Até à sua conclusão foram dedicados um esforço e dedicação diários enormes que desde início achei, em consciência, serem minha obrigação. Para a manutenção desse empenho em muito contribuíram um conjunto de pessoas que me acompanharam, directa ou indirectamente, no meu dia-a-dia.

Gostava de deixar um agradecimento ao meu orientador, Professor Carlos Silvestre, pela cordialidade e correcção com que acompanhou todo este processo, conduzindo esforços e traçando objectivos claros e ambiciosos. Obrigado pela oportunidade proporcionada de estar integrado no laboratório DSOR e na equipa fantástica que o constitui. Um agradecimento particular por ter criado a oportunidade de trabalhar, em paralelo com a tese, numa Bolsa de Iniciação Científica. Estendo o agradecimento ao Professor Paulo Oliveira, meu co-orientador de tese, pela frontalidade e espírito crítico que sempre demonstrou, bem como o esforço adicional que teve no acompanhamento e supervisão da fase final da tese.

Deixo uma palavra ao Bruno Cardeira, antigo aluno do IST e agora pertencente à equipa de investigação do DSOR, pelo acompanhamento incessante e motivador do meu trabalho, pelos conselhos sábios e experiência transmitida. Quero agradecer também aos restantes membros da equipa, especialmente à Rita Cunha e ao David Cabecinhas, que sempre me auxiliaram sem qualquer constrangimento.

À companhia indispensável dos meus amigos Henrique Silva, André Silva e Jorge Ribeiro, e aos muitos cafés que serviram para discussão de ideias, nas mais variadas áreas, muito obrigado!

Uma nota carinhosa à minha namorada por me ter acompanhado nesta última jornada, com muita paciência e espírito crítico, bem como a presença afectiva constante, indispensável para a conservação e preservação da minha boa disposição e concentração ao longo destes meses.

Um agradecimento muito especial aos meus pais e irmão, pelo apoio incansável e paciência inesgotáveis nos bons e maus momentos. Pela motivação e crença na minha capacidade de trabalho. Pela atenção e disponibilidade que sempre demonstraram. Pela dimensão que tiveram, ao longo do meu percurso académico e vida, nas minhas conquistas e sucessos. Muito, muito obrigado!



# Abstract

More and more Nature constitutes a source of inspiration in solving problems in the area of Autonomous Systems. Certain behaviours that happen spontaneously in Nature are the target of numerous investigation projects.

An exemplar case is the one of Unmanned Autonomous Vehicles (UAVs), since everything that is sought for those vehicles, in terms of navigation, is perfectly executed by insects, which have limited processing capabilities due to their biological characteristics.

In specific environments such as indoors, absolute positioning sensors - GPS for instance - can't be used and new alternatives should be investigated. This thesis aims to apply the biological concept of optical flow, which is the visual motion of features in an image, to the navigation of an UAV with specific characteristics, the quadrotor.

Firstly, quadrotor is introduced and depicted, through the study of kinematics and dynamics.

Afterwards, the way how optical flow can be used is described. Nowadays, the solution of complex optical systems with incorporated cameras and vision algorithms to obtain the information about motion is often used. The alternative suggested in this thesis is based on simple sensors with all the processing incorporated. These optical sensors are the basic component of an optical mice, commonly used by everyone on a daily basis. The presented work encompasses the study of the device and development of all communication with such a sensor, involving areas such as electronics and programming of microcontrollers. Given that these sensors are dimensioned for their use in optical mice, an optics study was needed to extrapolate its use to an aerial vehicle, with tremendously bigger working distances to the floor. A prototype was developed and installed on the quadrotor.

The last stage of this thesis comprised the use of the retrieved data, which needs compensation for rotational errors originated by quadrotor's angular rates. The data were included in a closed control loop for position stabilization - hover. The existing simulator was adapted to the needs of this project and a 3D model was developed for real-time visualization of the simulation results.

**Keywords:** Unmanned Aerial Vehicle, Quadrotor, Optical Flow, Optical Mice, Hover



# Resumo

Cada vez mais a Natureza surge como fonte de inspiração na resolução de problemas associados a sistemas autónomos. Determinados comportamentos que visam ser replicados em inúmeros projectos de investigação científica nesta área acontecem espontaneamente na Natureza.

O caso de veículos aéreos autónomos (UAV) é exemplar, já que tudo aquilo que se pretende para um veículo deste tipo, em termos de navegação, é executado na perfeição por insectos, com limitadas capacidades de processamento associados às suas características biológicas.

Em determinados ambientes, como dentro de um edifício, sensores de posicionamento absoluto - como por exemplo, o GPS - não podem ser utilizados, sendo que novas alternativas devem ser investigadas. A presente tese visa aplicar o conceito biológico de *optical flow*, que é a noção de movimento visual de uma imagem, à navegação de um veículo aéreo autónomo com características específicas, o *quadrotor*.

Primeiramente, o veículo em causa é introduzido, através do estudo da cinemática e dinâmica.

Posteriormente, a forma como o *optical flow* pode ser usado para navegação é descrita. Actualmente as formas mais correntes e usadas de cálculo de *optical flow* baseiam-se em sistemas ópticos complexos com câmaras incorporadas nos veículos e necessidade de processamento de imagem e visão. A alternativa sugerida nesta tese baseia-se em sensores com processamento incorporado. Estes sensores ópticos são a base de funcionamento dos ratos ópticos usados comumente no dia-a-dia. O trabalho a apresentar baseou-se então no estudo do dispositivo e no desenvolvimento de toda a comunicação com um sensor deste tipo, desde a electrónica, ao trabalho de programação de micro-controladores. Dado que estes sensores estão dimensionados para determinada funcionalidade num rato, um estudo óptico foi necessário para extrapolar este conceito para um veículo aéreo, que necessita de distâncias de funcionamento ao solo muito maiores. Um protótipo foi desenvolvido e instalado a bordo do *quadrotor*.

A última fase deste trabalho assentou no processamento de dados do sensor, através da compensação de efeitos rotacionais do movimento do *quadrotor*, e inclusão destes numa cadeia fechada de controlo do veículo para estabilização de posição - *hover*. O simulador existente foi adaptado às necessidades do projecto e um modelo 3D foi desenvolvido para visualização dos resultados em tempo real.

**Palavras Chave:** Veículo Autónomo Aéreo, Quadrotor, Optical Flow, Ratos Ópticos, Hover



# Contents

<b>Agradecimientos</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Resumo</b>	<b>v</b>
<b>Contents</b>	<b>v</b>
<b>Table of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Objectives and Motivation . . . . .	2
1.2 State of the Art . . . . .	2
1.3 Contributions . . . . .	3
1.4 Thesis' Organization . . . . .	4
<b>2 Quadrotor</b>	<b>5</b>
2.1 Quadrotor Kinematics . . . . .	5
2.1.1 Rotation Matrix . . . . .	5
2.1.2 Rotation Representation . . . . .	6
2.1.3 Kinematics . . . . .	9
2.2 Quadrotor Dynamics . . . . .	11
2.2.1 Rigid Body Dynamics . . . . .	11
2.2.2 Forces and Moments . . . . .	14
2.3 Aerial Platform . . . . .	16
<b>3 Optical Flow</b>	<b>17</b>
3.1 Concept . . . . .	17

3.1.1	Navigation and Flight Control . . . . .	19
3.2	Hardware . . . . .	19
3.2.1	Sensor - Avago ADNS5050 . . . . .	21
3.2.2	Lenses . . . . .	23
3.2.3	Interface . . . . .	25
3.2.4	Test Platform . . . . .	26
3.2.5	Prototype . . . . .	31
<b>4</b>	<b>Control</b>	<b>35</b>
4.1	Hover Control . . . . .	35
4.1.1	Optical Flow Compensation . . . . .	35
4.1.2	Simulator . . . . .	41
4.1.3	Optical Flow Controller . . . . .	41
4.2	3D Model . . . . .	47
4.3	Multiple Sensor Fusion . . . . .	47
<b>5</b>	<b>Conclusions and Future Work</b>	<b>49</b>
5.1	Future Work . . . . .	49
	<b>Appendices</b>	<b>57</b>
<b>A</b>	<b>Communicating with ADNS-5050</b>	<b>57</b>
<b>B</b>	<b>Compensation of rotational components</b>	<b>61</b>
<b>C</b>	<b>Root Locus</b>	<b>65</b>
<b>D</b>	<b>12 mm Lens vs 16 mm Lens</b>	<b>67</b>

# List of Figures

2.1	Rotation of a reference frame. . . . .	6
2.2	Inertial to body reference frame transformation. . . . .	7
2.3	Spherical center and four point masses model used for calculating the moment of inertia of the quadrotor. . . . .	12
2.4	On the left: perspective view of the quadrotor, with the positive rotations shown for pitch, roll and yaw, as well as the direction of rotation of the propellers. On the right: top view diagram of the quadrotor, with all the torques and forces that actuate on the frame of the vehicle. . . . .	14
2.5	IST Quadrotor. . . . .	16
3.1	Visual sensing of an insect. . . . .	18
3.2	Optical flow system present in optical mice. . . . .	20
3.3	(a) Avago ADNS5050. (b) Diagram of the sensor's chip. . . . .	21
3.4	Converting cpi displacement in radians, by means of the field of view of the sensor. . . . .	22
3.5	Refraction of light through a convex lens. $S_1$ is the distance to the focused object, $S_2$ is the distance from the sensor to the lens and $f$ is the focal distance. . . . .	24
3.6	Field of view ( $\alpha$ ) determination. $S_1$ is the distance to the focused object, $S_2$ is the distance from the sensor to the lens, $f$ is the focal distance and $d$ is the size of the CMOS array. . . . .	25
3.7	Field of view as a function of the focal distance $f$ . . . . .	26
3.8	Overview of the system's architecture. . . . .	27
3.9	(a) Calibration railway. (b) White high-intensity LED pointing towards the target. (c) Frame captured by the optical system at a distance of 50 cm of the target. . . . .	28
3.10	Test platform. . . . .	29
3.11	(a) SQUAL measured on the gym at a height of 30 cm. (b) SQUAL measured on the gym at a height of 50 cm. (c) SQUAL measured on the lab at a height of 30 cm. (d) SQUAL measured on the lab at a height of 50 cm. . . . .	30
3.12	(a) M-12 BT16020 lens. (b) Diagram of the experimental apparatus. (c) Alustar LED module. . . . .	31

3.13	(a) Raw displacement data at a sample rate of 5ms. (b) Position at a sample rate of 5ms. (c) Raw displacement data at a sample rate of 30ms. (d) Position at a sample rate of 30ms.	32
3.14	(a) SQUAL at a sample rate of 5ms. (b) SQUAL at a sample rate of 30ms. . . . .	33
3.15	Quadrotor with the sensor incorporated. . . . .	34
3.16	Electric wiring schematics of the sensor and the micro-controller. . . . .	34
4.1	Two distinct movements - translational and rotational - can result in the same optical flow measurement. . . . .	36
4.2	Difference between the quadrotor's body frame and the sensor reference frame. . . . .	37
4.3	Optical flow compensation test. . . . .	38
4.4	Linear path test. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position. . . . .	39
4.5	Oscillatory path test. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position. . . . .	40
4.6	Simulink quadrotor model. . . . .	41
4.7	Method for controlling the quadrotor. . . . .	42
4.8	Implemented proportional controller. . . . .	43
4.9	Simulation results for $ K_\theta ,  K_\phi  = 5$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position. . . . .	44
4.10	Simulation results for $ K_\theta ,  K_\phi  = 12$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position. . . . .	45
4.11	Simulation results for $ K_\theta ,  K_\phi  = 25$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position. . . . .	46
4.12	3D Model added to the simulator. . . . .	47
A.1	Write operation. . . . .	57
A.2	Read operation. . . . .	58
A.3	Operation's timings. . . . .	59
B.1	Optical flow compensation test. . . . .	61
B.2	Test for the compensation of rotational motions. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position. . . . .	62
C.1	Root locus of one of the transfer functions. . . . .	66

D.1 Test for the comparing two different lenses and their effectiveness. (a) Using a lens with  $f = 12$  mm. (b) Using a lens with  $f = 16$  mm. . . . . 67



# List of Tables

2.1	6-DOF Motion. . . . .	10
3.1	LED versus Laser characteristics. . . . .	20
3.2	ADNS5050 pin description. . . . .	21
3.3	Optical flow set-ups. . . . .	27
A.1	Timing restrictions of the communication protocol with ADNS-5050. . . . .	59



# Abbreviations

**AIRTICI** - Advanced Interactive Tools for the Inspection of Critical Infrastructures

**CAN** - Control Area Network

**CMOS** - Complementary Metal-Oxide-Semiconductor

**DOF** - Degrees Of Freedom

**DSOR** - Dynamical Systems and Ocean Robotics

**FOE** - Field Of Expansion

**GPS** - Global Positioning System

**IMU** - Inertial Measurement Unit

**INS** - Inertial Navigation System

**IST** - Instituto Superior Técnico

**OF** - Optical Flow

**SPI** - Serial Peripheral Interface

**UAV** - Unmanned Aerial Vehicle

**USART** - Universal Synchronous Aynchronous Receiver Transmitter



# Chapter 1

## Introduction

Robotics are a main subject in academic and industrial research nowadays. New projects and solutions appear in different areas. Human kind aims at creating totally autonomous agents able to perform and adapt to real problems that arise in natural environments.

Intelligent agents, so to speak, merely reflect the behaviours which were thought for the robot itself. By incorporating a set of sensors on a machine, the available data allows one to think, process and, finally, through a set of actuators, react accordingly.

A recently popular approach to robotics involves using biological behaviours that happen spontaneously in Nature. In artificial intelligence for instance, methods like neural networks aim to somehow replicate the learning process of a brain, by using training as a way to adjust mathematical models in artificial neurons in response to certain inputs and correspondent outputs. Instead of building simple agents, based on rules, the target is to create a learning process. Another example are genetic algorithms, which are search heuristics based on human evolution to optimize problems.

Different solutions have been applied from soccer robots to vacuum cleaners. The enormous diversity of applications is solely limited to human creativity and imagination. Indoor and outdoor oriented robots have been developed, from aerial vehicles such as air planes and helicopters, to water vehicles such as sail boats and submarines.

The state of the art in aerial robotics has moved rapidly from simple systems based on RC models, only able to do basic hover or cruise flights using inertial sensors, to robotic vehicles able to navigate and perform simple missions using GPS, vision sensors, lasers, among others.

Once again Nature comes up as a source of inspiration. Flies and insects are the closest thing to aerial vehicles. The manned aircraft developed over the past 100 years rely on a similar lift to the generated by birds' wings. Aircraft designers have paid little attention, however, to the pilot's visual sensor that finely controls these wings, although it is definitely the most sophisticated avionic sensors ever known to exist. The key issues arising in aerial navigation, like flying through cluttered environments and close to obstructions (obstacle avoidance, take-off and landing), are perfectly solved by flying insects. These are

agile creatures that navigate swiftly through most unpredictable environments, despite having limited brain processing capacity. Everything humans sought for in the field of aerial robotics is achieved by these creatures: tracking, docking, manoeuvres, collision avoidance, autonomous navigation, etc.

The last decades evidenced that insects guide themselves by processing the optical flow (OF) generated on their eyes as a consequence of their movement. Systems based on the same principle should be able to replicate the smoothness in insects' locomotion. By incorporating similar sensors on aerial vehicles, one expects to accomplish very positive results in the aforementioned fields of navigation.

## 1.1 Objectives and Motivation

Involved in the scope of the Advanced Interactive Robotic Tools for the Inspection of Critical Infrastructures (AIRTICI) project funded by AdI, the work reported in this document intended to use an inexpensive optic flow sensor, as the ones in optical mice, to measure translational displacements of an aerial platform, a quadrotor. The focus of this research is comprised of four main smaller objectives.

First step was to attempt to interface the sensor, developing all the necessary libraries to communicate and extract valid data.

The second step was to develop a navigation algorithm that would incorporate these experimental sensors to stabilize the vehicle in hover position and, later on, trying to extend the functionality of the sensor to manoeuvres like landing.

The third step was to use the available vehicle simulator to incorporate a simulated sensor and test the navigation algorithms.

The final step was to assemble the sensor on the vehicle and test it on an indoor environment.

The envisioned scenario for this system was for the quadrotor to be able to successfully maintain the hover position using optical flow on the university campus indoor gym.

## 1.2 State of the Art

Unmanned Aerial Vehicles (UAVs) and, specifically, quadrotors, have many important uses including forest fire surveillance, civilian search and rescue, border patrol, environmental and traffic monitoring, disaster relief, military applications, critical infrastructure analysis, mapping, etc. It is straightforward to notice in the existing literature that the research interests in control and navigation of UAVs has increased tremendously in recent years. This is related to the increasingly number of applications suitable for these types of vehicles, due to their versatility, complexity and portability. There are some disadvantages attached to these systems as well, namely their range, payload and power limitations. Nonetheless, from military interests to pure amusement purposes, numerous vehicles have been modelled and built.

Combined with their unique characteristics, these vehicles are also an excellent platform able to support a huge diversity of sensors, hence their extensive use in academic and industrial research. More

recently, optical flow has been a target of research applied aerial vehicles. Its association with the navigation of aerial vehicles has been emphasized by many people. [3] aims at proposing, analysing and demonstrating new ways of implementing optical flow sensors, based on relevant theoretical issues concerning the optical flow computation. Later on, [7] presented several research efforts aimed at developing new sensing and control algorithms inspired by insect vision and flight behaviours. [14] presents several autopilots, such as OCTAVE, LORA III and OSCAR, all bioinspired visual-motor control systems for the guidance of micro-air vehicles.

One major approach to the computation of optical flow involves the use of onboard cameras and processing capacity, such as the systems presented in [12], [11], [10] and [9]. Other approach is based on existing electronics to solve the computational side of optical flow, as a mainstream solution for optical mice commonly used. The applicability of some of these sensors has been studied by several people and has been intensely developed over the past few years. A deep study on these sensors is made by [13], where several tests are performed to the sensors, ending up in a very extensive set of results about the quality, effectiveness and usefulness of these sensors.

The application of these devices in the world of robotics and navigation is also real. Evaluation of a simple optical sensor has been made by [22], estimating the motion of a robot by visual means and examining the results for rough terrain environment, where odometry faces the dead reckoning problem.

A remote terrain navigation for unmanned air vehicles was presented by [2], where challenges associated with canyon and urban navigation missions were treated. OF based ranging was developed, to reactively avoid obstacles while flying along a preplanned path. *optiPilot* is a control strategy which can cope with take-off and landing using a small set of inexpensive optic flow sensors. Here, height above ground is estimated by fusing the data with other information. Analogue to this procedure, height above ground is also studied in

### 1.3 Contributions

The contributions of this thesis are mainly experimental. The main contributions include:

- The study, design and construction of prototype optical flow displacement sensor using the Agilent ADNS-5050 optical flow sensor;
- Theoretical overview of lenses and experimental validation on the prototype, for the purpose of extending the working height above the floor;
- Use of an optical flow sensor in a quadrotor, compensating the sensor data with IMU data, to extract both the translation and rotational components of displacements;
- Incorporation of a simple 3D-Model in the existing simulator; and
- Hover control using the sensor data and adaptation the existing controllers, tested in simulation.

## 1.4 Thesis' Organization

The thesis is organized so as to describe the the developed system and its performance. First, in Chapter 1, the quadrotor and the optical flow theme are described, with current investigation outcomes presented. In Chapter 2 the aerial platform is introduced and its model described, through the study of kinematics and dynamics. Chapter 3 is the gateway to optical flow. There, the whole concept is described and its application to navigation introduced. The sensor used for the thesis is studied and a new lens is chosen based on a small optics estimation. A final prototype is presented and the first results shown. The inclusion of the prototype in the vehicle and the use of data for hover control purposes is depicted on Chapter 4, where the existing simulator is presented and adapted, with the inclusion of a 3D model. Finally, Chapter 5 introduces the conclusions and future work.

# Chapter 2

## Quadrotor

This chapter will provide a useful introduction to the quadrotor used in this thesis.

The study of the quadrotor is mainly divided in two parts: *kinematics* (Section 2.1), which concerns only geometric aspects of motion, disregarding mass or forces, and *dynamics* (Section 2.2), which is the analysis of the forces that cause these motions.

### 2.1 Quadrotor Kinematics

Usually it is necessary to use different coordinate systems for several reasons ([6]):

- Forces and torques are applied in the body frame, i.e. the coordinate frame attached to the quadrotor, and so are Newton's equations of motion.
- The majority of sensors used in the vehicle provide information with respect to the same frame. An exception to this is a GPS sensor. Its measurements are with respect to the inertial frame.
- Most missions and objectives planned for these vehicles are based on trajectories and paths specified in the inertial frame. Additionally, map information is also described in the inertial frame.

The transformation between frames is given by two basic operations: rotations and translations. On Section 2.1.1 rotation matrix is introduced and Section 2.1.2 describes the usage of different representations for the orientation of the vehicle, namely the Euler angles and the rotation vector. Section 2.1.3 illustrates the quadrotor kinematics using the information contained on previous sections. Finally, the quadrotor dynamics is shown in Section 2.2 with detailed and intuitive explanation of the model used.

#### 2.1.1 Rotation Matrix

In order to explain the meaning of a rotation matrix, consider Fig.2.1, where  $\{F_1\}$  (specified by  $x_1$  and  $y_1$ ) is obtained by performing a right-hand rotation of  $\{F_0\}$  (specified by  $x_1$  and  $y_1$ ) by  $\theta$  radians. Vector  $\mathbf{p}$  can be described using either one of the referred reference frames.

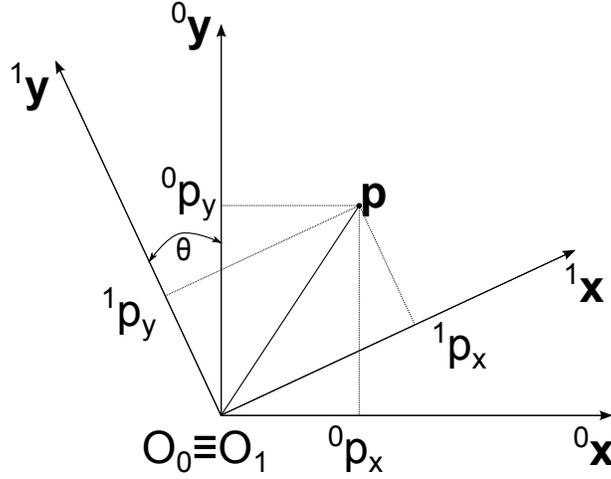


Figure 2.1: Rotation of a reference frame.

Using a rotation matrix, it is then possible to transform the coordinates of any vector  $\mathbf{p}$  from one reference frame to another. Considering the notation used in Fig.2.1, i.e.  ${}^i\mathbf{p}$  corresponds to vector  $\mathbf{p}$  described in the reference  $\{F_i\}$ , then one can make such transformation using the following equation:

$${}^1\mathbf{p} = {}^1_0\mathbf{R}^0\mathbf{p} \quad (2.1)$$

The square matrix  ${}^i_0\mathbf{R}$  transforms the coordinates of fixed vector  $\mathbf{p}$  from an expression in  $\{F_1\}$  to an expression in  $\{F_0\}$ , where the latter has been obtained by a right-hand rotation of  $\{F_0\}$ . Rotation matrices are orthogonal matrices, i.e. matrices whose columns and vectors are orthogonal unit vectors. As such, they enjoy a set of important properties:

- $({}^B_A\mathbf{R})^{-1} = ({}^B_A\mathbf{R})^T = {}^A_B\mathbf{R}$
- ${}^C_B\mathbf{R}{}^B_A\mathbf{R} = {}^C_A\mathbf{R}$
- $\det {}^B_A\mathbf{R} = 1$

The rotation matrix in Eq.2.1 might be interpreted in two different ways. In contrast to the one already mentioned, the second interpretation does not involve a change of frame. Instead, can represent a left-hand rotation of vector  $\mathbf{p}$  to a new vector in the same reference frame.

### 2.1.2 Rotation Representation

There are many ways of parametrizing rotations in three dimensions, namely Euler angles, rotation vector (or angle-axis) and quaternions ([5], [15]). Each parametrization has its own advantages and disadvantages, discussed in [4]. The first two will be described next.

### 2.1.2.1 Euler Angles

Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body. They constitute a means of representing a spatial orientation of any frame, through a composition of rotations from a frame of reference.

In this representation, rotations are made sequentially over the Euler angles, leaving the other two constant. After each partial rotation, the orientation of the original frame is changed. The popularity of such representation is attached to their physical meaning, since Euler angles constitute a direct match to the angles measured by onboard sensors of inertial navigation systems (INS).

It is important to mention two important reference frames for these platforms: inertial and body. Inertial reference frame  $\{I\}$  is one in which the motion of a particle not subject to forces is in a straight line at constant speed. Earth can be considered flat and non-rotating due to its size and the distance and duration of flights. The X-axis is chosen to lie in the Northward direction while Y lies East and Z points down. Hence this is an Earth fixed coordinate system. The body reference frame  $\{B\}$  is attached to the navigation platform, i.e. the quadrotor, and its origin is at the center of mass of the vehicle. However, the axes of  $\{B\}$  are aligned with the axis of  $\{I\}$ .

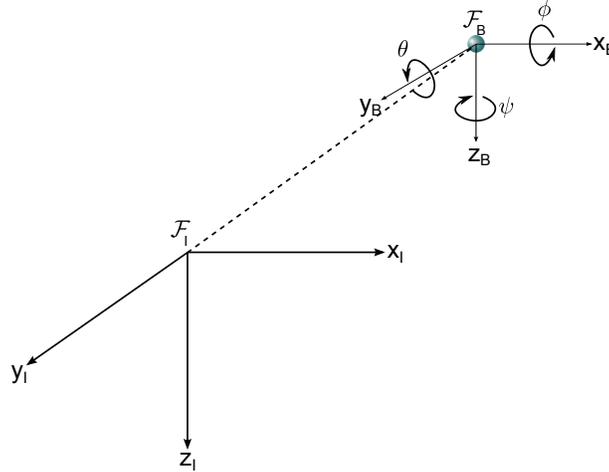


Figure 2.2: Inertial to body reference frame transformation.

The orientation or attitude of the rigid body, i.e. the quadrotor, is defined by  $\lambda = [\psi, \theta, \phi]'$ , where yaw ( $\psi$ ), pitch ( $\theta$ ) and roll ( $\phi$ ) describe rotations around Z, Y and X, respectively, as shown in Fig.2.2.

According to the properties mentioned in Section 2.1.1, when performing Z, Y and X rotations sequentially, the geometric transformation from  $\{I\}$  to  $\{B\}$  is a compose rotation and is given by

$${}^B_I \mathbf{R} = {}^B_{F_2} \mathbf{R}(\phi) {}^F_2 \mathbf{R}(\theta) {}^F_1 \mathbf{R}(\psi) \quad (2.2)$$

where  $\{F_1\}$  and  $\{F_2\}$  are intermediate frames. Following the inverse property

$${}^I_B \mathbf{R} = ({}^B_I \mathbf{R})^T = {}^I_{F_1} \mathbf{R}(\psi) {}^F_2 \mathbf{R}(\theta) {}^F_2 \mathbf{R}(\phi) \quad (2.3)$$

Replacing each rotation by the corresponding matrix gives Eq.2.4

$${}^I_B \mathbf{R} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.4)$$

Finally, the final rotation matrix is obtained by performing the product in Eq.2.4.

$${}^I_B \mathbf{R} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.5)$$

where  $c \equiv \cos$  and  $s \equiv \sin$  from now on, for simplification. There are many possible representations using Euler angles [16], depending on the order in which one performs the rotations. The one described so far is named Z-Y-X, for obvious reasons.

It is possible to obtain the Euler angles Z-Y-X through the entries of the rotation matrix. Considering the generic matrix

$${}^I_B \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.6)$$

The Euler angles can be obtained by combining Eq.2.6 and Eq.2.5. Using the trigonometric properties and crossing entries from both matrices

$$\begin{aligned} \theta &= \arctan2\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \\ \psi &= \arctan2\left(\frac{r_{21}}{c\theta}, \frac{r_{11}}{c\theta}\right) \\ \phi &= \arctan2\left(\frac{r_{32}}{c\theta}, \frac{r_{33}}{c\theta}\right) \end{aligned} \quad (2.7)$$

Singularities exist in Eq.2.7. Solutions according to a convention used in [17] might be chosen.

### 2.1.1.2 Rotation Vector (Angle-Axis)

This parametrization is totally characterized by a vector  $\boldsymbol{\varphi}$  which describes a rotation in three dimensions using only its module (an angle  $\Lambda$ ) and its direction (unit-vector  $[\varphi_x \varphi_y \varphi_z]$ ). It is a compact

notation, with three elements.

$$\boldsymbol{\varphi} = \Lambda \begin{bmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} \quad (2.8)$$

According to [19], the rotation matrix can be obtained by the rotation vector by the following expression

$$\mathbf{R} = \frac{\boldsymbol{\varphi}\boldsymbol{\varphi}^T}{\Lambda}(1 - \cos \Lambda) + I \cos \Lambda + \frac{\sin \Lambda}{\Lambda}\mathbf{S}(\boldsymbol{\varphi}) \quad (2.9)$$

Both  $\boldsymbol{\varphi}$  and  $\mathbf{R}$  refer to the same transformation (initial and final reference frame). The indexes were omitted for the sake of simplicity.  $\mathbf{S}(\cdot)$  stands for a skew matrix and represents the external product of a vector, as in Eq.2.10, for a general vector  $\mathbf{v}$ .

$$\mathbf{S}(\mathbf{v}) = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \quad (2.10)$$

The inverse calculation is also possible, simply developing Eq.2.9.

$$\Lambda = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right) \quad (2.11)$$

$$\begin{bmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} = \frac{1}{2 \sin \Lambda} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$$

Finally, it is also important to refer the dynamics of this parametrization, deduced in [19]

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} + \frac{1}{2}\mathbf{S}(\boldsymbol{\varphi})\boldsymbol{\omega} + \frac{1}{\Lambda^2}\left(1 - \frac{\Lambda \sin \Lambda}{2(1 - \cos \Lambda)}\right)\mathbf{S}(\boldsymbol{\varphi})\mathbf{S}(\boldsymbol{\varphi})\boldsymbol{\omega} \quad (2.12)$$

usually named Bortz's equation, which describes the time derivative of the rotation vector. There are certain issues with this notation, since there are singularities in Eq.2.11, for  $\Lambda = 0$  and  $\Lambda = \pi$ .

### 2.1.3 Kinematics

The expressions derived in this chapter are general to any rigid body, but the notation and coordinate frames used are specific to the quadrotor.

Consider  $\mathbf{p} = [x, y, z]^T$  as the quadrotor's position in  $\{I\}$ ,  $\mathbf{v} = [u, v, w]^T$  as the linear velocity of the vehicle expressed in  $\{B\}$ ,  $\boldsymbol{\lambda} = [\theta, \phi, \psi]^T$  as the orientation given by the euler angles with respect to  $\{F_1\}$ ,  $\{F_2\}$  and  $\{I\}$ , respectively, and  $\boldsymbol{\omega} = [p, q, r]^T$  as the angular velocity expressed in  $\{B\}$ .

The kinematics of this vehicle relate state variables on different coordinate frames. Since 6 independent coordinates are necessary to determine the position and orientation in space, the relationships between them are described with 6 DOF (degree of freedom) kinematic equations. Each motion DOF is an entry of Table 2.1.

DOF	Linear		
1	motion in the $x$ -axis direction	$u$	$x$
2	motion in the $y$ -axis direction	$v$	$y$
3	motion in the $z$ -axis direction	$w$	$z$
DOF	Angular		
4	rotation about the $x$ -axis (roll)	$p$	$\phi$
5	rotation about the $y$ -axis (pitch)	$q$	$\theta$
6	rotation about the $z$ -axis (yaw)	$r$	$\psi$

Table 2.1: 6-DOF Motion.

Considering the state variables  $x$ ,  $y$  and  $z$  all refer to the inertial frame and  $u$ ,  $v$  and  $w$  to the body frame, it is fairly intuitive to conclude that

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^I_B \mathbf{R} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.13)$$

However, when trying to establish a relationship between the absolute angles  $\phi$ ,  $\theta$  and  $\psi$  and the angular rates  $p$ ,  $q$  and  $r$ , the task is much more complicated. The difficulty arises due to the fact that while the latter are defined in the body frame,  $\phi$ ,  $\theta$  and  $\psi$  are all defined in different frames,  $\{F_2\}$ ,  $\{F_1\}$  and  $\{I\}$ , respectively. As such

$$\boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + {}^B_{F_2} \mathbf{R} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + {}^B_{F_2} \mathbf{R} {}^{F_2}_{F_1} \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.14)$$

Using the transpose matrices of the ones present in Eq.2.4

$$\begin{aligned} \boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ \boldsymbol{\omega} = \mathbf{T}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\theta & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \quad (2.15)$$

Inverting the matrix

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{T} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec(\theta) & c\phi \sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.16)$$

Note once again that Euler angles derivatives are defined in different reference frames. Although they can be written in vector form for algebraic purposes, they aren't a vector in  $\mathbb{R}^3$ .

## 2.2 Quadrotor Dynamics

Having introduced the motion of the vehicle, it is important to study what causes movement: the dynamics.

### 2.2.1 Rigid Body Dynamics

The forces and moments respect the laws of Newton, which can only be held in inertial frames. Assuming the dynamics for a rigid body, Newton's second law applied to translational motion and the equation of Coriolis give

$$m \frac{d\mathbf{v}}{dt_i} = m \left( \frac{d\mathbf{v}}{dt_b} + \boldsymbol{\omega} \times \mathbf{v} \right) = \mathbf{f} \quad (2.17)$$

Since the force is applied in  $\{B\}$ , it is possible to express Eq.2.17 in body coordinates, where  $\mathbf{v} = [u, v, w]^T$ ,  $\boldsymbol{\omega} = [p, q, r]^T$  and  $\mathbf{f} = [f_x, f_y, f_z]^T$ . Therefore

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (2.18)$$

The same procedure can be adopted to rotational motion, where force is equivalent to torque and linear momentum to angular momentum.

$$\mathbf{J} \frac{d\boldsymbol{\omega}}{dt_i} = \boldsymbol{\tau} \quad (2.19)$$

and, applying the Coriolis equation

$$\mathbf{J} \frac{d\boldsymbol{\omega}}{dt_i} = \mathbf{J} \frac{d\boldsymbol{\omega}}{dt_b} + \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega}) = \boldsymbol{\tau} \quad (2.20)$$

where  $\mathbf{J}$  is the constant inertia matrix and  $\boldsymbol{\tau}$  is the applied torque.

$\mathbf{J}$  describes the dynamic behavior of a body in rotating around a defined axis. It is solely dependent on physical properties, as is mass.

The constant inertia matrix can then be described as

$$\mathbf{J} = \begin{bmatrix} J_{XX} & J_{XY} & J_{XZ} \\ J_{YX} & J_{YY} & J_{YZ} \\ J_{ZX} & J_{ZY} & J_{ZZ} \end{bmatrix} \quad (2.21)$$

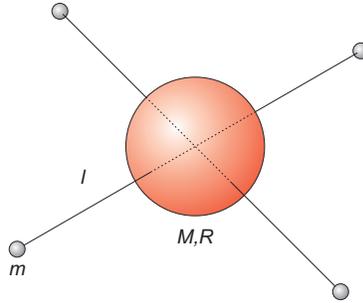


Figure 2.3: Spherical center and four point masses model used for calculating the moment of inertia of the quadrotor.

The model adopted for the quadrotor includes a spherical dense center with mass  $M$  and radius  $R$ , with point masses of mass  $m$  located at a distance of  $l$  from the center.

Theoretically, symmetry exists about all three axis. Therefore, it is reasonable to assume that  $\mathbf{J}$  is a diagonal matrix, since  $\mathbf{J}_{XY} = \mathbf{J}_{YX} = \mathbf{J}_{XZ} = \mathbf{J}_{ZX} = \mathbf{J}_{YZ} = \mathbf{J}_{ZY} = 0$ . Using linear algebra, it is trivial to calculate its inverse.

From [6], the moment of inertia of a sphere is well known, given by  $J_{ii} = \frac{2MR^2}{5}$ ,  $i = \{X, Y, Z\}$ . Also, [1] states that, when the center of mass of the object does not coincide with the axes origin (coincident with the center of gravity of the quadrotor), like the four point masses of Fig.2.3, the moment of inertia can be computed using the parallel axis theorem. If the object with mass  $M$  is displaced by a distance  $D$

from the axis of rotation, then the moment of inertia of the displaced object can be calculated according to the following equation

$$J_{displaced} = J_{center} + MD^2 \quad (2.22)$$

Point masses have no volume, so the contribution while rotating around themselves -  $J_{center}$  - is null. Therefore, from Fig.2.3 it is possible to conclude that

$$\begin{aligned} J_{XX} &= \frac{2MR^2}{5} + 2ml^2 \\ J_{YY} &= \frac{2MR^2}{5} + 2ml^2 \\ J_{ZZ} &= \frac{2MR^2}{5} + 4ml^2 \end{aligned} \quad (2.23)$$

Note that the moment of inertia for axis  $X$  and  $Y$  only reflect the contribution of the two point masses not aligned with the respective axis (otherwise, the distance  $D$  is obviously zero).

Writing Eq.2.20 in body coordinates and defining  $\boldsymbol{\tau} = [\tau_\phi, \tau_\theta, \tau_\psi]^T$ , plus solving it with respect to  $\frac{d\boldsymbol{\omega}}{dt}$  gives

$$\frac{d\boldsymbol{\omega}}{dt} = \mathbf{J}^{-1} [\mathbf{S}(\boldsymbol{\omega})\mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}] = \begin{bmatrix} \frac{J_{YY} - J_{ZZ}}{J_{XX}} qr \\ \frac{J_{ZZ} - J_{XX}}{J_{YY}} pr \\ \frac{J_{XX} - J_{YY}}{J_{ZZ}} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{XX}} \tau_\phi \\ \frac{1}{J_{YY}} \tau_\theta \\ \frac{1}{J_{ZZ}} \tau_\psi \end{bmatrix} \quad (2.24)$$

Besides the force exerted by the motors, the quadrotor is also under the influence of gravity. Therefore, the gravitational component of the total force exerted on the vehicle must also be encompassed on the model. This force, applied to the center of mass of the vehicle, can be described on the inertial frame by

$${}^I \mathbf{F}_g = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.25)$$

Considering that Eq.2.18 is written in  $\{B\}$ , the vector must be transformed to the same frame, holding

$${}^B \mathbf{F}_g = {}^B \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} -mg \sin(\theta) \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix} \quad (2.26)$$

Finally, the total model is obtained by grouping Eqs.2.13,2.16,2.20,2.24

$$\begin{aligned}
 \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} &= {}^I_B \mathbf{R} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
 \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} &= \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -mg \sin(\theta) \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix} + \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \\
 \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi \sec(\theta) & c\phi \sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\
 \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} \frac{J_{YY} - J_{ZZ}}{J_{XX}} qr \\ \frac{J_{ZZ} - J_{XX}}{J_{YY}} pr \\ \frac{J_{XX} - J_{YY}}{J_{ZZ}} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{XX}} \tau_\phi \\ \frac{1}{J_{YY}} \tau_\theta \\ \frac{1}{J_{ZZ}} \tau_\psi \end{bmatrix}
 \end{aligned} \tag{2.27}$$

### 2.2.2 Forces and Moments

A more general approach is commonly necessary in order to better understand how to maneuver the quadrotor. It has four identical rotors and the propellers have a fixed angle of attack. The opposing rotors are paired and each pair rotates in a different direction. In other words, two of the propellers rotate in a clockwise direction and the other two spin counter-clockwise.

Each propeller produces a force, usually named thrust, and a torque, both of which are proportional to the square of speed of rotation.

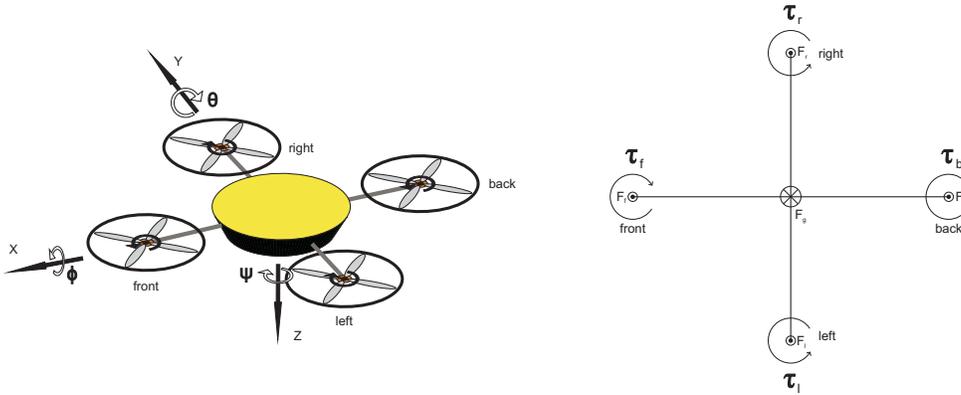


Figure 2.4: On the left: perspective view of the quadrotor, with the positive rotations shown for pitch, roll and yaw, as well as the direction of rotation of the propellers. On the right: top view diagram of the quadrotor, with all the torques and forces that actuate on the frame of the vehicle.

Newton's Third Law, which expresses the relation between the mutual forces of action and reaction between two bodies, is what governs the motion of the quadrotor. The propellers apply a force to the air accelerating it downwards. The reaction of the air applies an equal and opposite force on the propellers accelerating them upwards. The same can be transposed to torques: motors exert a torque to the propellers and the propellers provide an equal and opposite torque to the motor. Note that the motors are rigidly attached to the frame of the quadrotor. Thus, the reaction torque is applied directly to the frame. This description is depicted on Fig.2.4.

There are four non-redundant control inputs, meaning that the quadrotor is an underactuated vehicle. These are:

- **Thrust -  $F$**

The thrust input results from the sum of the forces produced by each of the rotors, so that

$$\mathbf{T} = \mathbf{F}_f + \mathbf{F}_b + \mathbf{F}_l + \mathbf{F}_r \quad (2.28)$$

In the body frame, the total force acting on the vehicle is given by

$$\mathbf{F} = {}^B_I \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (2.29)$$

- **Roll Torque -  $\tau_\phi$**

Torque inputs result from the force differential between two rotors on the same axis.. To create a positive roll torque, the left propeller's speed should increase while the right one is decreased by the same amount. On the contrary, if the intended goal is a negative torque, the right one should increase while the left one is decreased.

$$\tau_\phi = l(F_l - F_r) \quad (2.30)$$

- **Pitch Torque -  $\tau_\theta$**

Similarly, the pitching torque is produced by balancing the forces of the front and back motors, as follows

$$\tau_\theta = l(F_f - F_b) \quad (2.31)$$

- **Yaw Torque -  $\tau_\psi$**

As described previously, the yawing torque is controlled by all the motors. The net thrust and pitch/roll torques can be balanced if the equalities of Eq.2.28,2.31,2.30 are kept constant, i.e.,

increasing (decreasing) the remaining motors' speed by the same amount while decreasing (increasing) the rest in the same proportion. This creates a positive (negative) yaw torque, making the quadrotor turn around itself.

$$\tau_\psi = c(F_f + F_b - F_l - F_r) \quad (2.32)$$

In summary, the thrust and torque inputs are given by

$$\begin{bmatrix} T \\ \tau_\theta \\ \tau_\phi \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ c & c & -c & -c \end{bmatrix} \begin{bmatrix} F_f \\ F_r \\ F_b \\ F_l \end{bmatrix} \quad (2.33)$$

## 2.3 Aerial Platform

The quadrotor used for this project is shown in Fig.2.5. It has a PC104 AEWIN PM6100A-G with a 500MHz AMD processor, 512MB RAM, IDE Compact Flash, RS232 ports, USB ports and Ethernet 10/100Mbps interface. A peripheral is used to provide a single CAN interface.

It has four AXI 4120/18 Gold Line brushless motors , hardened by a steel shaft.

There are several sensors which were supported by this vehicle, namely a pressure sensor, a GPS receiver, laser range finder and a nano IMU (inertial measurement unit).

It weighs 5.132 Kg and its arm is 0.41 m. The inertia matrix, obtained from a 3-D model of the vehicle designed in SolidWorks®, is given by:

$$\begin{bmatrix} 1.926e-1 & -2.643e-3 & 3.237e-5 \\ 2.643e-3 & 1.898e-1 & 5.539e-4 \\ 3.237e-5 & 5.539e-4 & 3.468e-1 \end{bmatrix}$$



Figure 2.5: IST Quadrotor.

# Chapter 3

## Optical Flow

This chapter introduces the whole subject of investigation, which is optical flow present in optical mice.

Firstly, a more abstract introduction is made, describing the concept of optical flow and the way it might be used in an UAV's navigation algorithm. Secondly, a mouse sensor is presented and described. Approaches concerning optical parameters are depicted and the final prototype is presented. Finally, the inclusion of this technology in the quadrotor is explained.

### 3.1 Concept

The dimensions of an aircraft pose a constraint to the desired missions for the vehicle. Like for the quadrotor, many proposed UAV applications include flying low over the ground and between obstacles. This requires navigation based on collision avoidance and altitude control based on sensing other than GPS. The aggressive mission environment also poses different challenges, like occlusions. These types of missions also constrain the size of the vehicle. One needs to provide the capability of weight reduction of sensors so that they can fit on smaller vehicles, by reducing the size of traditional avionics sensors.

Typical sensors for navigation are standard inertial measurement units (IMUs), pressure sensors for altitude and GPS for absolute positioning. Nowadays, the use of radars and laser range finders is also spreading. The mentioned sensors are appropriate for larger vehicles that generally fly at higher heights above ground. Although the electronics will eventually lead to a decrease in the size of these devices and, consequently, allow fitting them in a smaller UAV, there are disadvantages to be considered, namely concerning GPS. Signal jamming and update rates are factors which would eventually cause the vehicle to fail while flying in ever-changing unknown environments and close to the earth's surface.

Naturally, insects come as source of inspiration, which despite having low processing capabilities, have developed light and robust sensors and control systems, through natural evolution. As small brains limit the complexity of image processing, a vehicle is also limited due to its size and the characteristics of the

used hardware architecture, namely the processing speed. Flying insects spend their adult life in complex environments with a huge variety of obstacles, some of which they land on and take off from.

Not only insects make heavy use of visual clues. Humans also control their movements mainly resorting to the information received from their eyes. This visual sensing gives perception of depth and movement. These cues are provided by "optical flow" or visual motion.

Optical flow refers to the apparent movement of textures in the visual field of the insect, while moving. This information can be used to perceive depth and estimate the proximity to objects. On the one side, while looking at the downward direction, the insect estimates its speed from the observed optical flow. Faster optical flow indicates lower altitudes (assuming the insect's velocity doesn't change much). On the other side, obstacle detection can be made by detecting expansion or divergence in the visual field. This is depicted on Fig.3.1, where a small representation of an insect moving towards a tree is shown.

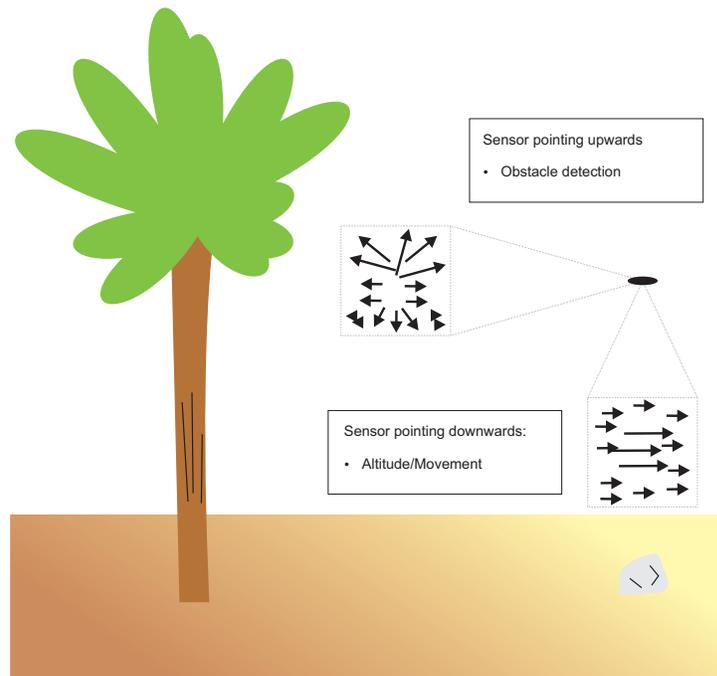


Figure 3.1: Visual sensing of an insect.

The downward view is useful for tracking purposes and terrain following. Notice the sudden increase of the field while passing above a stone in the ground.

More rapid expansions in the forward view imply a closer proximity to the obstacle. Usually the origin of the field is called the focus of expansion (FOE) and, if located inside a rapidly expanding region, a collision is imminent. On the contrary, if the FOE is located outside the expanding region, then the insect will move without colliding. These are the principles behind the behaviours of a flying insect, using optical flow.

### 3.1.1 Navigation and Flight Control

The question is now how insects use the optical flow field to navigate and control the flight in a stable way. [7] describes a study made and states that the adopted behaviours based on the information received by the sensors are quite simple. Tasks are performed using stratagems which can be seen as reflexes, making them appropriate for control loops.

Some of the described stratagems are:

- **Centering Response** - Two fly a collision-free path between two objects, equalize the optical flow measured on both sides. This heuristic ensures that, for instance, inside a tunnel, an insect flies at the center, maintaining the same distance to the left and right walls.
- **Landing Strategy** - Keep the downwards optical flow constant and the forward speed proportional to the vertical speed. Decreasing the forward speed proportional to the height guarantees a constant downwards optical flow and perfect landing, since when the height approaches zero, so does the flight speed.
- **Hovering Strategy** - Cancelling the image velocity everywhere allows the insect to hover. With zero velocity relative to the objects, the optical flow field is null. If the objects are stopped, so is the insect.

These are some of the reasons why biomimetic visual sensing can be applied to avionics.

The measurement of optical flow is usually done using a camera. It is a popular subject of investigation, known to be a computationally intensive problem, like many of image processing algorithms. Only in the late few years processors have become fast enough to overcome the overload issues, not jeopardizing the efficiency of the system with the time taken to compute the algorithms.

Another way of calculating the flow has recently popped up, by using electronic microsensors available in optical mice.

## 3.2 Hardware

Computers have evolved a lot since their origin. Attached to the machine itself, peripherals also sustained a large improvement over the years.

This is definitely the case with mice. The most general technology nowadays comprehends the use of optical sensors in mice. These sensors are mainly optical flow sensors.

These systems are usually defined by a complete image processing system, encapsulated in a tiny chip, represented in Fig.3.2. It is composed by three main elements:

- **Image Acquisition System** - just like a camera, it is constituted by an array of light sensors, which individually transform light intensity in an electric signal

- **Digital Signal Processor** - calculation of optical flow values from the acquired images
- **Serial Communication Interface** - responsible for establishing a serial communication with other devices

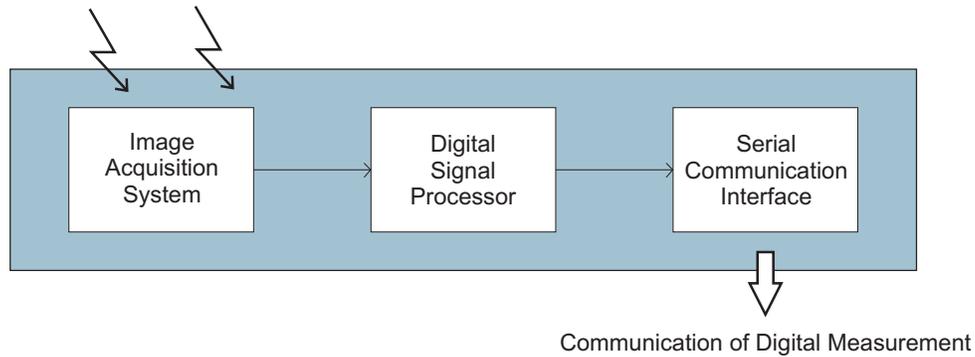


Figure 3.2: Optical flow system present in optical mice.

Associated with these sensors are usually two elements: a lens and a light source. The objective of these elements is to improve the performance of the sensor.

Lenses for optical mice are usually made of plastic and built-in in a support visible on the lower part of the mouse.

Concerning the light source, it can be of two types: light-emitting diode (LED) or laser. Since the movement detected is calculated by tracking distinctive features in the image, the key is obviously a good contrast. Hence, the surface must reflect enough light and it must have a random texture or pattern. Imagine someone is taking pictures or making a movie, while moving. Assume that the camera is pointing downwards, vertically. If the floor has random textures or patterns, consecutive images will have different features and, as such, the person’s movement will be detected in the sequence of acquired images. If, on the contrary, there is a repetitive pattern, a sequence of images will most likely be ambiguous, in the sense that there are similar features and no information about the direction of the movement. The reliable image correlation computation requires both high-contrast image and randomness.

Light Source	Characteristics
LED	Spontaneous emission, random phase, spread of wavelenghts
Laser	Stimulated emission, nearly single wavelength (keeps wavelike shape over time and distance)

Table 3.1: LED versus Laser characteristics.

One can state that while the LED light is somehow "incoherent", the laser is the opposite. This constitutes the fundamental advantage of the laser over LED.

### 3.2.1 Sensor - Avago ADNS5050

Avago's ADNS5050 (IC) is commonly used in optical mice. It is a cheap, common and lightweight device, based on a CMOS technology for capturing images, at a maximum rate of 4500 frames per second. As mentioned earlier, its theory of operation is based on the system shown in Fig.3.2. It measures changes in position (displacement) by acquiring sequential surface images and determining the direction and magnitude of movement, using internal procedures.

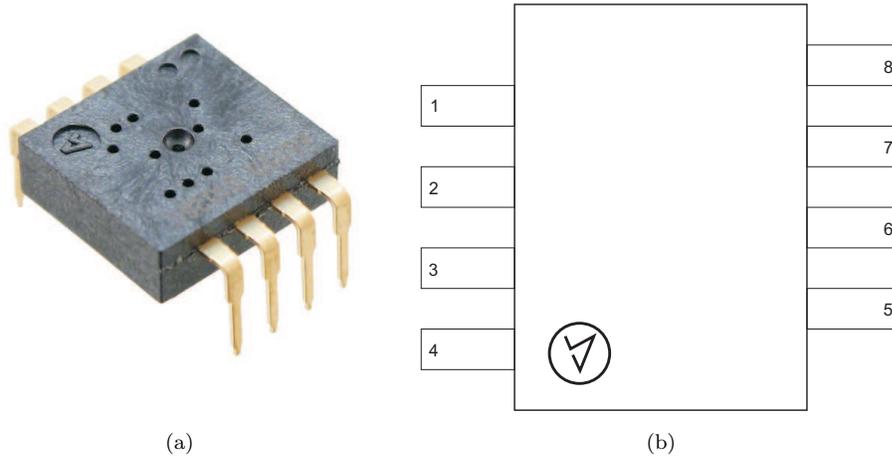


Figure 3.3: (a) Avago ADNS5050. (b) Diagram of the sensor's chip.

Pin	Name	Description	I/O Type
1	SDIO	Serial Port Data Input and Output	Input/Output
2	XY_LED	LED Control	Output
3	NRESET	Reset Pin	Input
4	NCS	Chip Select	Input
5	$V_{DD}$	Supply Voltage	Power
6	GND	Ground	Ground
7	Rego	Regulator Output	Output
8	SCLK	Serial Clock Input	Input

Table 3.2: ADNS5050 pin description.

Electronically, the sensor is programmed via registers through a three-wire Serial Peripheral Interface (SPI). It has a set of internal registers which can be read from or written to, through a synchronized transmission over the SDIO line. The synchronization is obtained with the clock line, shared between

the sensor and the other devices.

The sensor allows one to extract several types of data, among others:

- Displacement both in  $x$  and  $y$  coordinates;
- Measure of image quality, in terms of features, named SQUAL;
- Minimum and maximum light intensity in the image; and
- Frame captured by the sensor.

As the internal registers are all 8-bit, the extracted values are in the interval  $[-128, 127]$ , since they are represented in two's complement.

The optical flow provided by the sensor in both  $x$  and  $y$  directions is a single value. This means that there is no information about the optical flow field of the frame taken at a certain instant, but rather a mean value of the field in every pixel.

It is important to understand what information about the displacements is extracted. Displacements are measured in cpi (counts per pixel), i.e., the amount of pixels that, in average, the traceable features moved. The sensor accumulates measurements until a read operation is made and the internal registers are cleaned. For control purpose, the optical flow can be converted into a concrete physical value, other than cpi. For this, the notion of field of view has to be introduced. It corresponds to the angular extent of the observable world seen by an optical sensor, whether it is an eye, a camera or any other type of device. On Fig.3.4, it corresponds to the angular opening which encompasses the whole square image. The field of view can be seen individually for each dimension: there is an horizontal field of view and a vertical field of view, whether one considers x-axis or y-axis.

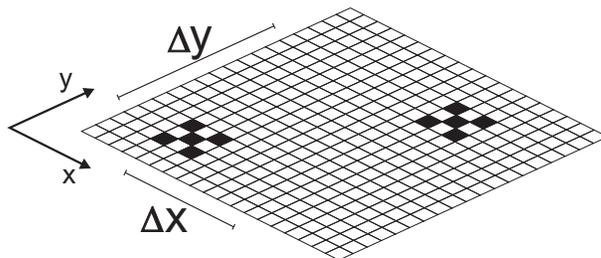


Figure 3.4: Converting cpi displacement in radians, by means of the field of view of the sensor.

Assuming a measured displacement in both directions between two time intervals, given by  $\Delta x$  and  $\Delta y$ , the conversion to radians is made by simple applying the following expression

$$\theta_i = \frac{\Delta i}{n_p} \alpha \tag{3.1}$$

where  $i$  is the defined direction ( $x$  or  $y$ ) and  $\alpha$  is the field of view in that direction.  $n_p$  is the total number of pixels in the desired direction. The angular movement of the sensor correspondent to the detected feature movement in one direction is a fraction of the total field of view in that same direction. Knowing the latter, the first can be determined by the values given by the sensor.

The ADNS5050 has been mainly designed for optical mice. It is dimensioned for close range applications, in the sense that it requires proximity to the surface in order to function properly. This is the reason why mice only work when dragged in a table. As soon as the device is raised above the table, it stops working, since there is a decrease on the reflected light and tables are usually somehow homogeneous in terms of features. As such, as soon as one detaches the sensor from the table, features aren't traceable any more.

The image captured by the ADNS5050 has 19x19 pixels. The limitation of its resolution constrains the applicability of the device. As one pixel retains the average of light intensity received, if the area captured by a single pixel is relatively large, features won't be tracked, since the image will be totally blurred. The light sensor incorporated in this integrated circuit is no more than an array of CMOS cells. Hence, the number of pixels is determined by the number of cells that constitute the array.

Considering the objective of controlling a quadrotor, it is obvious that this device isn't enough to obtain useful data. The use of this kind of sensors for navigation of an UAV require extending the potential of measuring the optical flow, by enlarging the working distance. This can be made by simply using a set of lenses, just like in a camera.

Instead of trial-and-error - an approach often used to define the characteristics of the desired lens - a small study of optical concepts was made. The idea was to be able to estimate the field of view based on the characteristics of the lenses.

### 3.2.2 Lenses

One of the most important components of an optical system is the lens, by means of which light rays are refracted so that they converge or diverge to form an image, depending if the lens is convex or concave. The refraction obeys to Snell's law of refraction, which relates the angles of incidence and refraction. When a light ray crosses an interface between two media of different refractive indices,  $n_1$  and  $n_2$ , the angle of incidence  $\theta_1$  and the angle of refraction  $\theta_2$  respect

$$\sin \theta_1 n_1 = \sin \theta_2 n_2 \tag{3.2}$$

This is what happens when a lens is used. Light is refracted twice: when it reaches the lens and when it exits the lens.

The principal axis of a lens is the imaginary line passing through the optical center. Light is neither reflected nor refracted when coincident with this axis.

A convex lens concentrates the rays of light parallel to the principal axis in a point named as focal point. The distance from the lens to this point is the focal distance  $f$ . It is one of the main characteristics of a lens, usually divulged by the manufacturer. It is a measure of the lens' curvature.

The image is formed through the intersection of every light rays originated from a real object that intersect the lens, as depicted in Fig.3.5.

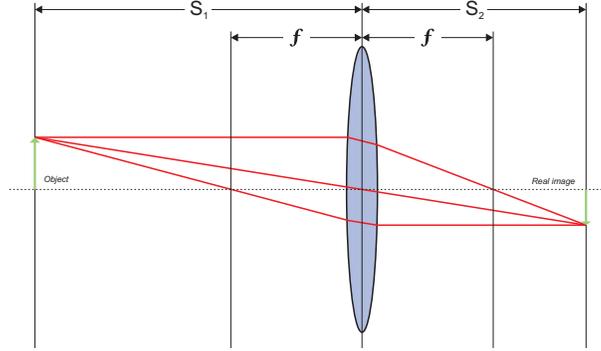


Figure 3.5: Refraction of light through a convex lens.  $S_1$  is the distance to the focused object,  $S_2$  is the distance from the sensor to the lens and  $f$  is the focal distance.

The thin lens equation is valid when the thickness of the lens is much more smaller than the focal distance. It comes from triangle equalities and relates  $S_1$  and  $S_2$  from Fig.3.5.

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f} \quad (3.3)$$

Both the focal distance and the physical dimensions of the sensor influence the final value of the field of view. The calculation of the field of view of an optical system is simple, as shown in Fig.3.6.

Applying simple trigonometric equalities

$$\tan\left(\frac{\alpha}{2}\right) = \frac{d}{2S_2} \quad (3.4)$$

Solving eq.3.3 with respect to  $S_2$  gives

$$S_2 = \frac{S_1 f}{S_1 - f} \quad (3.5)$$

Finally, applying this result to eq.3.4

$$\alpha = 2 \arctan\left(\frac{d(S_1 - f)}{2S_1 f}\right) \quad (3.6)$$

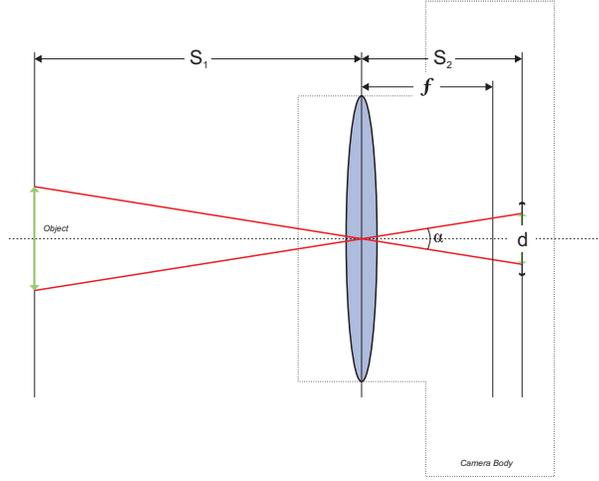


Figure 3.6: Field of view ( $\alpha$ ) determination.  $S_1$  is the distance to the focused object,  $S_2$  is the distance from the sensor to the lens,  $f$  is the focal distance and  $d$  is the size of the CMOS array.

A rough approximation can be done to simplify this expression. As  $S_1 \gg f$

$$\alpha \approx 2 \arctan \left( \frac{d}{2f} \right) \quad (3.7)$$

It is then possible to calculate the field of view using either the distance from the lens to the sensor or the desired ideal focus distance. Both implicitly depend on the focal distance. The bigger the focal distance, the smaller the field of view is. As one increases the distance from the lens to the sensor, the amount of light rays received decreases.

Knowing the distance to the object also allows to calculate its size, using the field of view. This is useful in the sense that it allows one to somehow estimate the size of traceable features.

The edge of the square seen by an array element is then calculated using the following expression

$$l_{edge} = 2 \tan \left( \frac{\alpha}{2} \right) d_{object} \quad (3.8)$$

where  $d_{object}$  is the distance from the lens to the object.

### 3.2.3 Interface

The interface with the sensor was achieved with an ATMEL90CAN128, a microcontroller from AT-MEL. It is a low-power CMOS 8-bit microcontroller. It has a considerable number of features like the use of general purpose I/O registers, timers, two USARTs and an SPI serial port.

The microcontroller was programmed to periodically extract data from the sensor.

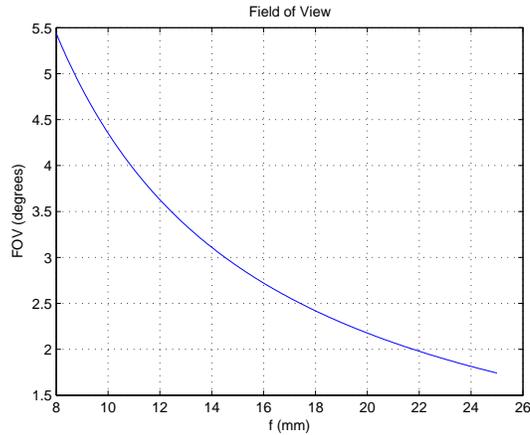


Figure 3.7: Field of view as a function of the focal distance  $f$ .

The SPI bus specifies two data lines besides the clock and the selection line - Master Output/Slave Input and Master Input/Slave Output - commonly referred to as MOSI and MISO, instead of the single one presented by the sensor. This presents a problem, as the sensor doesn't possess enough channels to communicate directly through dedicated hardware, supported by the microcontroller.

In order to communicate, the program running in the microcontroller has to be responsible for the direct control of the communication lines (SDIO, SCLK and NCS), as well as the required timings between clock changes. This software solution is called bit banging. In this, all the synchronization, changes and sampling of the communication lines have to be contemplated on the program running in the microcontroller. It is slower than other hardware solutions (usually interrupt-driven) since it is based on an active wait (processor is blocked until something is transmitted/received). It is also more susceptible to noise in the lines.

The protocol necessary to successfully communicate with the sensor is present in [20] and in Appendix A.

### 3.2.4 Test Platform

The know-how in optical flow has been extended over the latest few years. Before moving into a particular optical system, a research on existing systems was needed. Hence, three of these are referenced in Tab.3.3.

Pelican quadrotor, from Chemnitz UT, aims at using optical flow as odometer fused with Microsoft Kinect for indoors navigation, with position and velocity control. In Brigham Young University a flying wing airframe was developed and uses optical flow outdoors, for high altitude flights. Another approach - called Optipilot - uses these sensors for landing and take-off control of a swinglet. Despite the existing major differences, this set-up was firstly adopted in this project.

Vehicle	Type	Optical Sensor	Resolution	Rate	Lens
Pelican	Quadrotor	ADNS-3080	30x30	6400	M-12 Mount
Optipilot	Swinglet	ADNS-5050	19x19	4500	Collimator CAX-100
BYU	Flying Wing Airframe	ADNS-2610	18x18	1500	Kogaku

Table 3.3: Optical flow set-ups.

The development of a test platform was divided in a sequence of steps: communication, calibration of the lens and extraction and validation of the data.

### 3.2.4.1 Communication

Firstly, based on the sensor’s manual and the existing C library, a set of drivers was developed, in order to establish communication with the sensor. Being a passive device, it awaits for a trigger, usually set by the micro-controller. The debugging on micro-controllers is tricky, since there is no access to a console or monitor. There is no way of checking if the program flow is correct, unless an oscilloscope is used.

Secondly, there was the need to check data, namely the displacements, SQUAL and the current frame. For that purpose a MATLAB interface was developed, which receives data in real time. It is specially useful while calibrating, since one can visualize the captured image.

The architectural overview of the system can be seen on Fig.3.8.

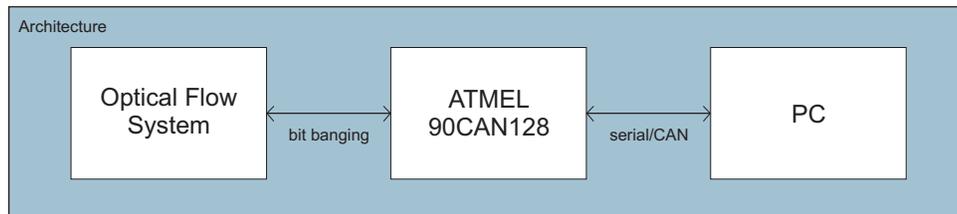


Figure 3.8: Overview of the system’s architecture.

The communication between the optical flow system (ADNS5050 and lenses) and the microcontroller is achieved through a bit banging solution, whereas the communication of the latter with a computer is made through either serial communication (MATLAB) or CAN (PC-104).

Data flows periodically from the sensor to the microcontroller, which is then responsible for forwarding it to a computer for control purposes.

### 3.2.4.2 Calibration

As mentioned in this chapter, these sensors are dimensioned for their use in an optical mice. It is necessary to extend the functionality of these devices by using lenses. The preliminary optical system was then composed by the ADNS5050 sensor and a 10 mm focal distance Collimator CAX-100 lens, just like Optipilot. As in any other optical system, calibration is required. Calibration in this case is adjusting the lens position and orientation, in order to have a reasonably focused image. To do so, a wooden railway was fixed to a table, directed to a white A4 paper sheet with a printed 'A'. This is shown in Fig.3.9.

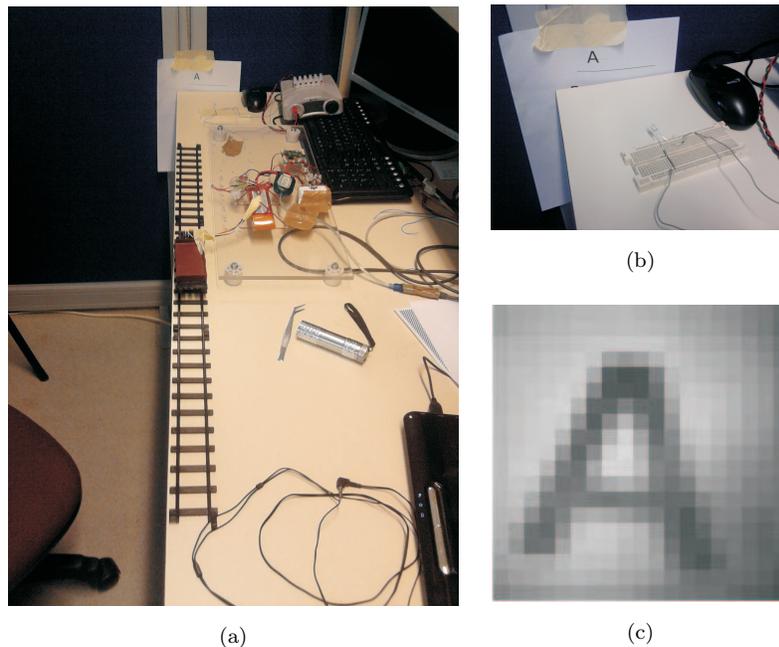


Figure 3.9: (a) Calibration railway. (b) White high-intensity LED pointing towards the target. (c) Frame captured by the optical system at a distance of 50 cm of the target.

For this process a white high-intensity LED was used, pointing directly to the target in the paper sheet, improving the contrast of the image taken by the sensor (see Fig.3.9). Later on, a red high-intensity LED was incorporated in the optical system, which was then fixed to a rigid plastic mount and used for data extraction. The final test platform consisted of the following elements, depicted on Fig.3.10.

- Electronic card with the ATMEL90CAN128;
- Electric Switch to control the power of the system;
- 12V Lipo (Lithium-ion polymer) battery with a low level voltage detector;
- ADNS5050 with an attached LED and a Collimax lens; and
- 1-Watt Resistor in series with the LED;

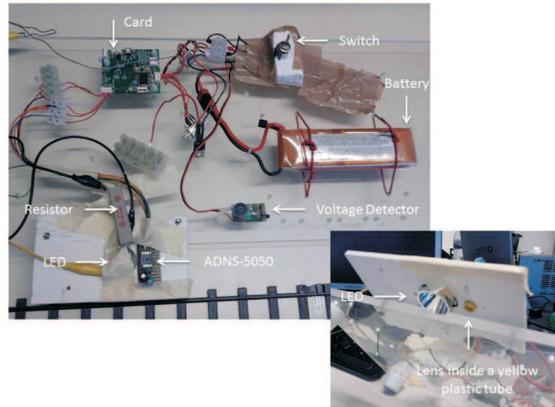


Figure 3.10: Test platform.

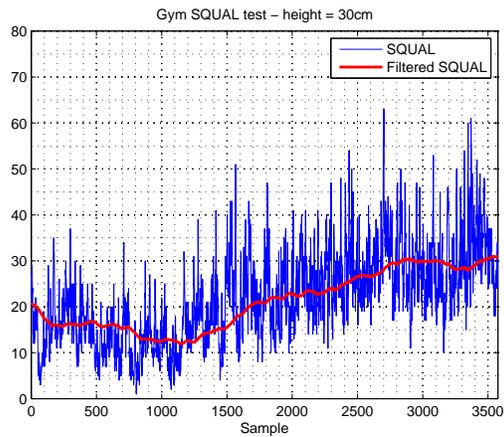
### 3.2.4.3 Data Extraction and Validation

Given the platform presented in the previous sections, a set of elementary tests was done in two different scenarios, contemplating distinct levels of features. The first was a gym's dark brown floor and the second was a bright laboratory floor, with tiny features (around 1 mm each). For each test, a fixed distance was covered by the sensor, approximately in a straight line, pointing downwards, vertically. The test was repeated at two different heights: 30 and 50 centimetres.

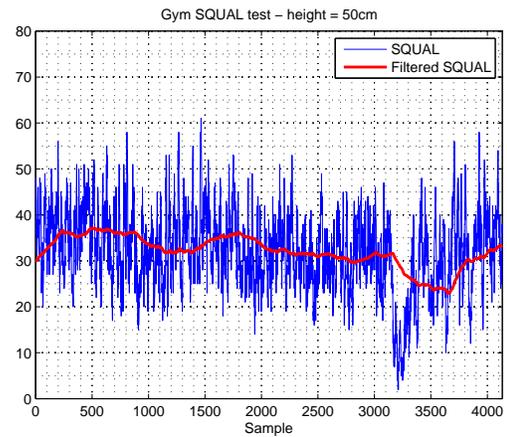
SQUAL results are presented in Fig.3.11. Both the raw data (in blue) and the low-pass filtered data (500-samples window average) are presented. The retrieved data from the gym presents a bigger oscillation due to the fact that the test was very inaccurate as the platform was hand-moved. In the laboratory, a wheeled support was used to support the system. This justifies the big difference in smoothness, outlined by the shown plots.

On the one hand, the dark brown floor has bigger features and, as such, they are more useful at higher distances from the floor. On the other hand, the laboratory's bright floor shows better results for smaller heights, since the features are very small. As the distance to the floor increases, the capability of tracking distinguishable features in the images decreases. Above 50 cm, data started to lose coherency. SQUAL values were very low compared to the ones presented. Recall Fig.3.7, where the field of view is given in respect to the focal distance. With  $f = 10mm$ , the field of view is of approximately  $4.4^\circ$ , which means that the the image taken by the sensor, at a distance of 50cm, is a square with an edge of 3.8 cm. This result can be achieved by using Eq.3.7. The experimental measurements of the calibration sheet, in Fig.3.10, indicated a 4x4 cm square, which is close to the theoretical value presented.

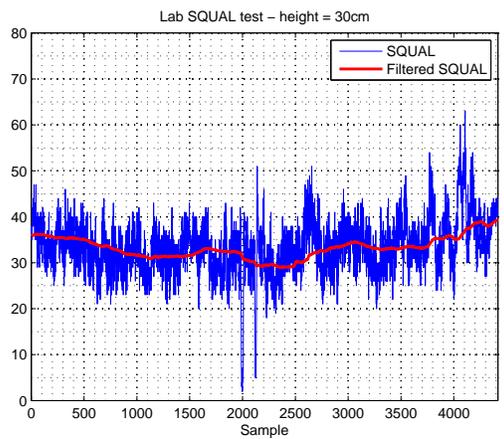
With the limited 19x19 pixel resolution, each pixel performs the average of an area of  $\frac{4}{19} \times \frac{4}{19}$  cm. The bigger the distance to the target, the bigger the area covered. Resolution is lost and image gets blurred.



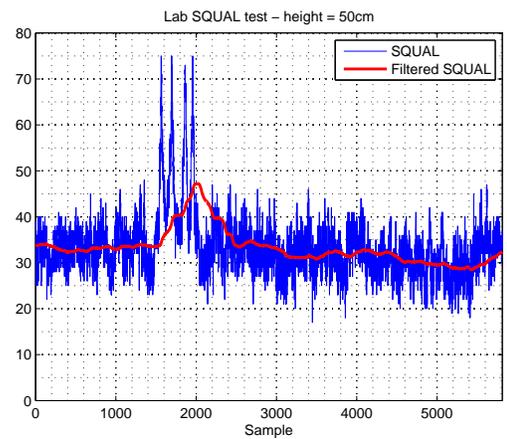
(a)



(b)



(c)



(d)

Figure 3.11: (a) SQUAL measured on the gym at a height of 30 cm. (b) SQUAL measured on the gym at a height of 50 cm. (c) SQUAL measured on the lab at a height of 30 cm. (d) SQUAL measured on the lab at a height of 50 cm.

Although very primary and elementary, these tests were vital to conclude that the optical system used didn't meet the needs of the project. In fact, using sensors at a working distance of 50 cm isn't secure at all for the quadrotor. The arm's length itself is almost that size.

This set-up was successful for the Optipilot mainly because the scope of that project aimed at outdoor flights, with no light issues (the daylight is most of the times enough) and a close proximity to the ground, imposed by the intended manoeuvring. Flights made by BUY's flying wing, for instance, worked at higher altitudes but the image features are considerably bigger, like trees, buildings, rocks, among others.

The objective of indoor flights combined with a desired ambitious distance from the floor wouldn't certainly be accomplished using the test platform.

### 3.2.5 Prototype

A new lens was required, to increase the working distance. To decrease the field of view and, consequently, increase the magnification (zoom) of the image, the focal distance should be bigger. However, this means that the amount of light that is captured is reduced and these sensors require a big amount of light to properly work. The increase of the focal distance should then account for this.

A set of lens, with a huge variety of focal distances, is available at Lensation. These M-12 mount lenses are ideal to be used in these projects, as they are lightweight and small. To comply with the necessary requirements, a new lens was incorporated in the optical system - a BT16020 ( $f = 16mm$ ) - with a proper adapter. Additionally, a red 1W Alustar LED module was also acquired. This module has an array of high-intensity red LEDs and incorporates the mount and mirror. Both components were placed in the test platform, replacing the old ones, and they can be seen in Fig.3.12.

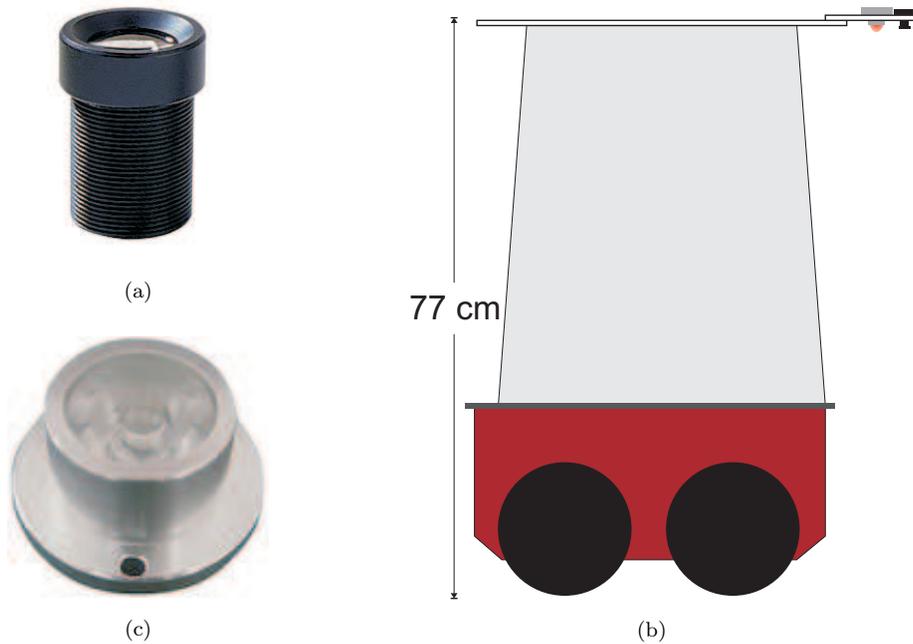


Figure 3.12: (a) M-12 BT16020 lens. (b) Diagram of the experimental apparatus. (c) Alustar LED module.

To validate the new optical system, the platform was assembled on the top of a terrain robot, a Pioneer 3-AT.

The program running in the robot consisted in a forward straight movement followed by the reverse movement. The initial and final position should coincide. Two different sample rates were tested (5ms and 30 ms), to check its influence on the extracted measurements. The sensor's height was bigger than in the previous tests.

For each sample rate, the data withdrawn from the sensor included the displacements and SQUAL value. Considering that the robot moved on a straight line and that the x-axis of the sensor was aligned with the direction of movement, only the displacements in that direction are relevant. As the internal registers are reset after a read operation, each displacement is relative to the last one made. In order to estimate the position relative to a starting point, consecutive displacements should be added. Hence, the position in  $i$ -axis of the vehicle at time instant  $k$  is given by

$$p_i(k) = \int_0^k \Delta i(k) dk, \quad i = x, y \quad (3.9)$$

The displacements in  $x$  and the position obtained from Eq.3.9 are shown for both sample rates.

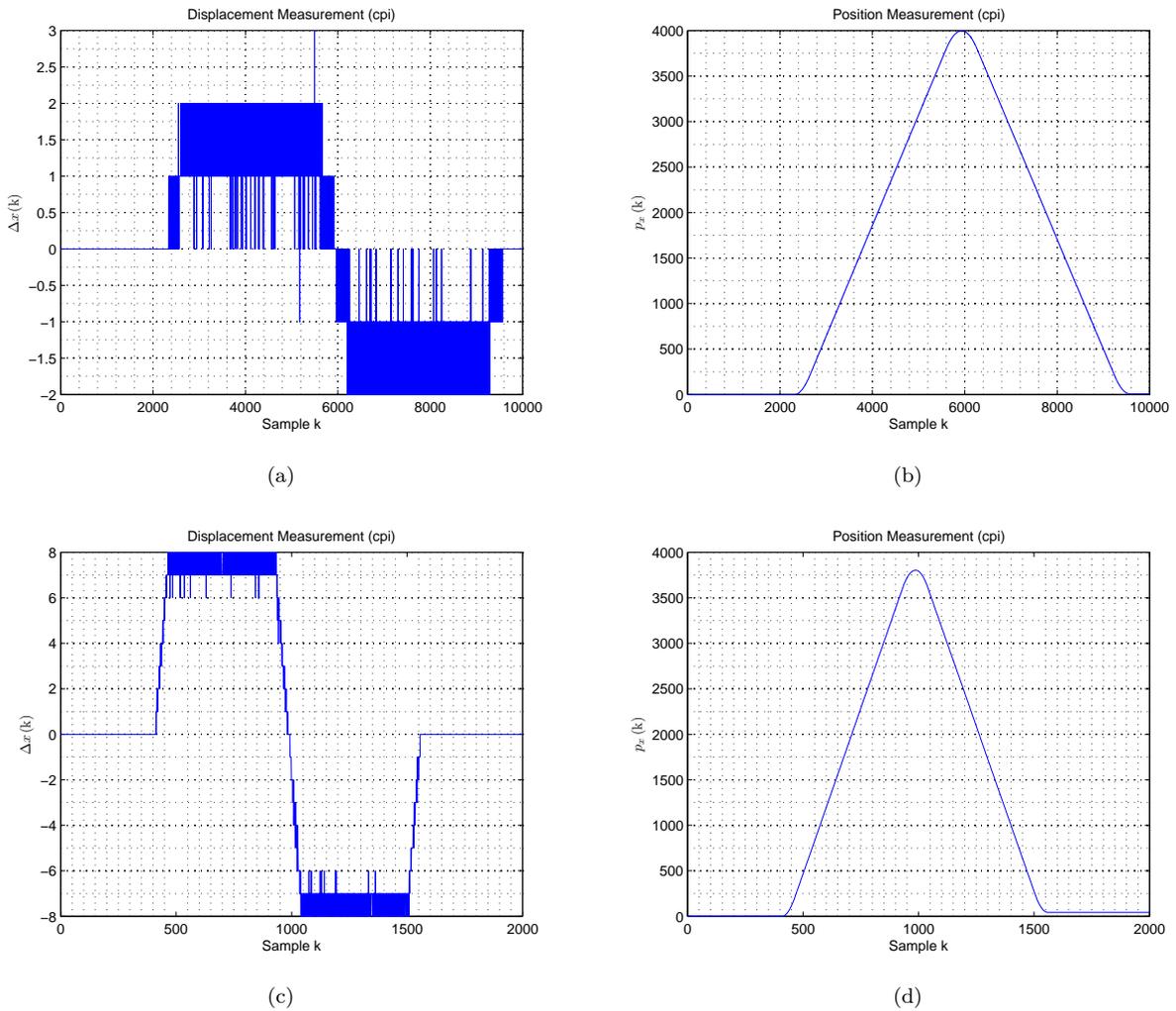


Figure 3.13: (a) Raw displacement data at a sample rate of 5ms. (b) Position at a sample rate of 5ms. (c) Raw displacement data at a sample rate of 30ms. (d) Position at a sample rate of 30ms.

Despite the sampling rate difference, the outcome of the experiment was practically the same. In terms of quality, both tests had very good results of SQUAL over time, as presented in Fig.3.14.

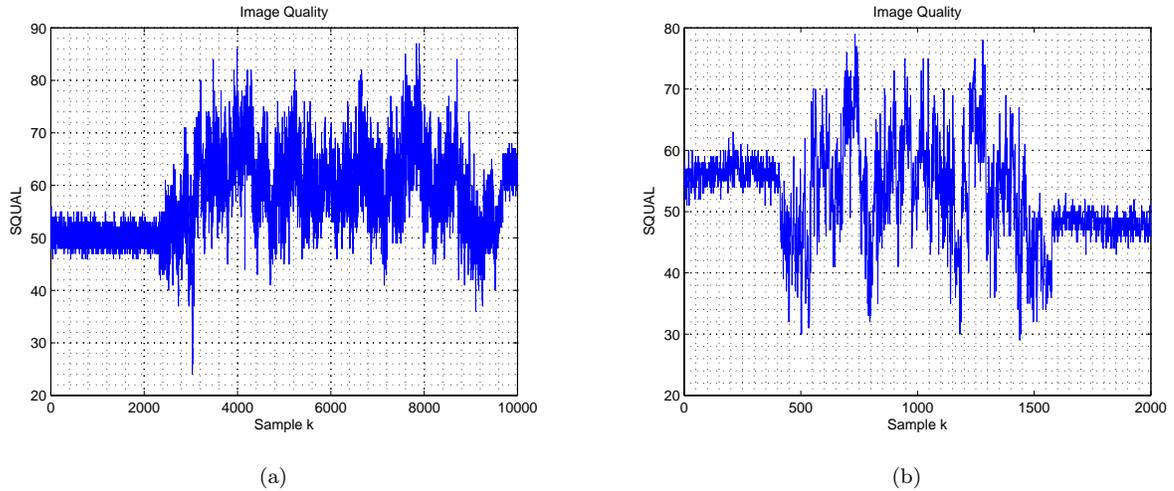


Figure 3.14: (a) SQUAL at a sample rate of 5ms. (b) SQUAL at a sample rate of 30ms.

As expected, the SQUAL value isn't influenced by the sampling rate. It solely depends on the quality of the image, in terms of features' amount and size. The change from the test platform to this last set-up contributed for an increase of the average SQUAL value, at higher distances from the floor.

It is also true that the lighting conditions are of utmost importance. When the LED isn't pointing towards the area targeted by the sensors, the SQUAL value decreases radically and the displacements accumulate a considerable error. By taking a look at an optical mouse, it is straightforward to understand the importance of light, since there should be a reasoning behind the fact that the red LED is fully concentrated below the mouse.

Given the frequencies involved in the communication between the micro-controller and the sensor, the wiring wasn't neither long nor thick enough to prevent signal corruption. A small touch in the wiring would cause a malfunction of the system. This is a frequent problem in accessing sensors with these types of data transmission, due to the reflection of the signal. The solution is to use a set of resistors at the end of the wiring, on the sensor side, in order to impedance match.

These results were the gateway to the final prototype. In order to assemble the optical system in the quadrotor, a new plastic support was made and the power wiring adapted to the set of batteries of the vehicle. A schematics of the whole wiring is shown in Fig.3.16. The assembled sensor can also be seen in Fig.3.15. The prototype is lightweight, so there isn't any dynamical constraint related to its inclusion in the vehicle's frame.

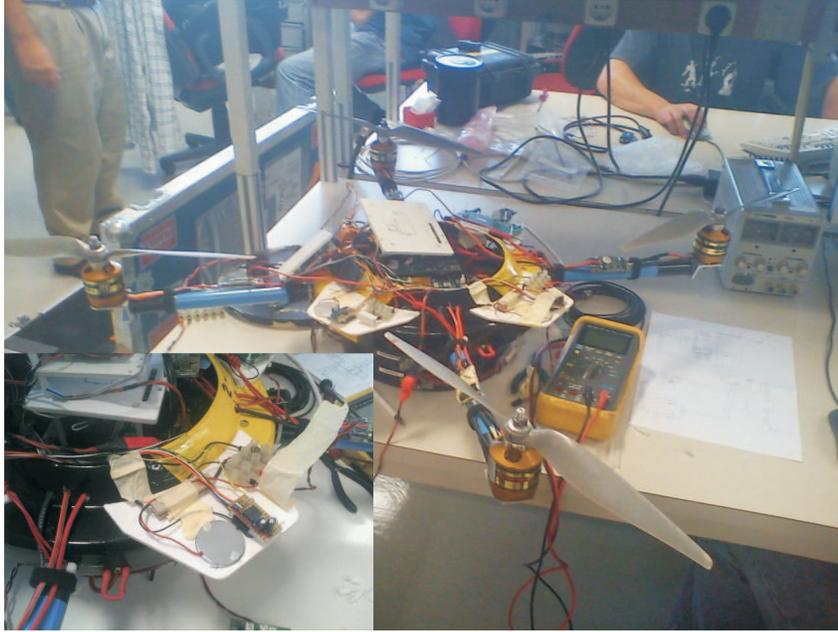


Figure 3.15: Quadrotor with the sensor incorporated.

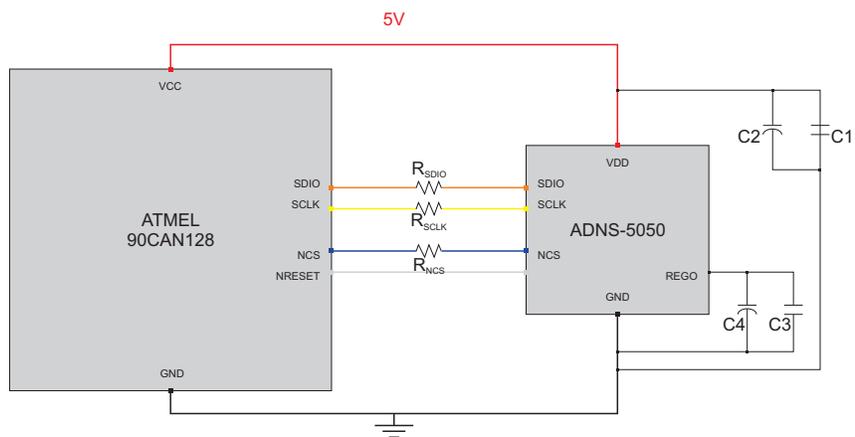


Figure 3.16: Electric wiring schematics of the sensor and the micro-controller.

# Chapter 4

## Control

The objective of this chapter is to demonstrate how the data from an optical flow sensor can be used to control the quadrotor. Section 4.1.1 indicates how optical flow data can account for attitude errors. In Section 4.1.2, an existing simulator for the quadrotor is presented and its main components described. Based on that simulator, Section 4.1.3 explains how to incorporate the sensor's data in a feedback loop in order to control the quadrotor, adjusting the existing simulator. A 3D-Model is presented, as a very good framework for debugging and tuning of control parameters. Finally, a simple approach for multiple sensors data fusion is also proposed.

### 4.1 Hover Control

Hovering is one of the most challenging processes when controlling a helicopter, such as the quadrotor. The air flux originated by the rotors act against the vehicle's frame, so there is a constant need for adjusting and correcting the control inputs to keep the hovering position.

Given the degrees of freedom of the quadrotor, the optical flow measured by the sensor will have two components: translational and rotational. For hover control, i.e., immobilizing the quadrotor in the air, only the translation component is important. Small rotational movements should have no impact on the response of the vehicle.

#### 4.1.1 Optical Flow Compensation

The measured optical flow can then be described as

$$\Delta i = \Delta i_{translational} + \delta i_{error} \quad (4.1)$$

where  $i = x, y$ .

While the translational component is due to the linear movement of the vehicle relative to the ground, the error comes from angular rates. Given that the sensor is attached to the frame of the quadrotor,

rotations around the  $x$ -axis and  $y$ -axis influence the measured optical flow.

The translational component is directly related to the linear velocity of the vehicle, parallel to the ground. Although the obtained displacement is a measure of motion, it is ambiguous about absolute velocity. The sensor in fact measures a ratio between velocity and height above ground. Consider an angular displacement  $\theta$  as the one described in Fig. 3.4, obtained by converting sensor data into radians. Applying simple trigonometric equations, the translational component is given by

$$\theta = \frac{v_g \Delta t}{h_{ag}} \quad (4.2)$$

where  $v_g$  is the velocity relative to the ground,  $\Delta t$  is the sampling time and  $h_{ag}$  is the height above ground. The same translational optical flow can be obtained for different velocities, as long as the ratio between velocity and height remains the same.

Besides the ambiguity in velocity, there is also an ambiguity related to the type of movement of the vehicle.

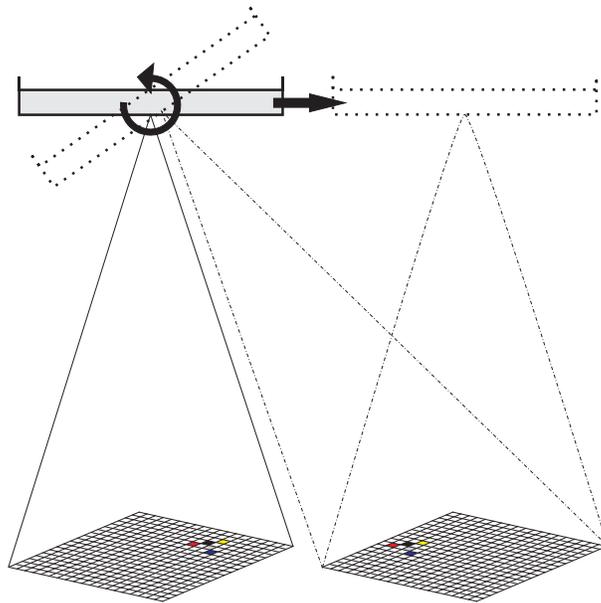


Figure 4.1: Two distinct movements - translational and rotational - can result in the same optical flow measurement.

In Fig.4.1, it is clear how the same optical flow can be obtained by two completely distinct manoeuvres. A rotation can get similar results as an horizontal displacement. However, as the objective is to control the position of the quadrotor, maintaining hover, one has to separate both components and use that information in the feedback loop.

The rotational motion can be effectively compensated for using the angular velocity  $\omega$  measured by the inertial measurement unit (IMU). One possible approach for this problem is mentioned in [18], where

the time-derivative of the rotation vector is used to calculate that component of optical flow, as follows

$$\begin{cases} \delta x_{error} = \left(\frac{n_p}{\alpha}\right) \dot{\varphi}_y \Delta t \\ \delta y_{error} = \left(\frac{n_p}{\alpha}\right) \dot{\varphi}_x \Delta t \end{cases} \quad (4.3)$$

where  $n_p$  is the data read from ADNS5050,  $\alpha$  is the field of view of the optical system and  $\Delta t$  is the sampling period.

Recall the calculation of this derivative, from Section 2.1.2.2. As stated previously, this approach has a very critical problem, related to the singularities that exist when the euler angles are close to zero. This happens very often in a quadrotor flight, which constitutes a problem for the proposed objective of correcting the rotational component of optical flow.

Alternatively, the method chosen for solving the compensation problem was to directly use the information given by the IMU and fuse it with the optical flow data. Similarly to Eq.4.3, the error added by rotational movements is given by

$$\begin{cases} \delta x_{error} = \left(\frac{n_p}{\alpha}\right) \omega_y \Delta t \\ \delta y_{error} = \left(\frac{n_p}{\alpha}\right) \omega_x \Delta t \end{cases} \quad (4.4)$$

Note that the IMU data is given in the body frame. The sensor isn't aligned with the body frame for structural issues, so the information can't be fused without transforming one frame into another. This can be done in a simple way. Consider a generic angle  $\beta$ , which corresponds to the angular shift between the sensor reference frame and the quadrotor body frame, as depicted in Fig.4.2.

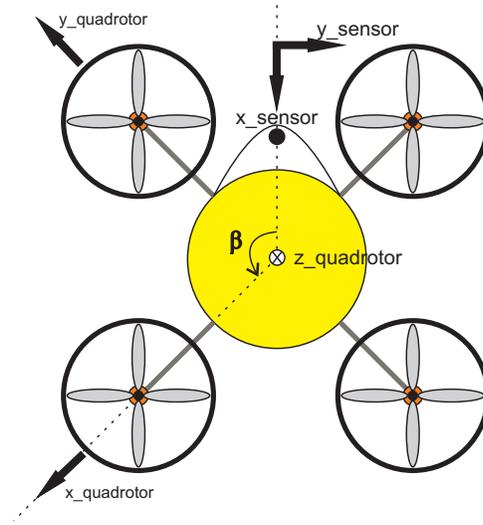


Figure 4.2: Difference between the quadrotor's body frame and the sensor reference frame.

The transformation of the displacement data is then given by two rotations. First, a rotation of  $\pi$

radians around  $y$  is made, which implies a signal change in the measured  $x$ -displacements. Afterwards, a rotation of  $\beta$  radians around  $z$  completes the transformation.

$$\begin{bmatrix} {}^B\Delta x \\ {}^B\Delta y \\ 0 \end{bmatrix} = \mathbf{R}_z(\beta)\mathbf{R}_y(\pi) \begin{bmatrix} {}^S\Delta x \\ {}^S\Delta y \\ 0 \end{bmatrix} \quad (4.5)$$

After this transformation, the data extracted from the sensor is aligned with the body frame of the quadrotor. All the calculations are now coherent, since they are relative to the same reference frame.

To validate the aforementioned corrections, two tests were performed on Instituto Superior Técnico's campus. The experiment is depicted on Fig.4.3.

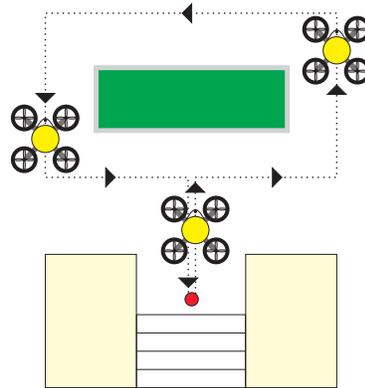


Figure 4.3: Optical flow compensation test.

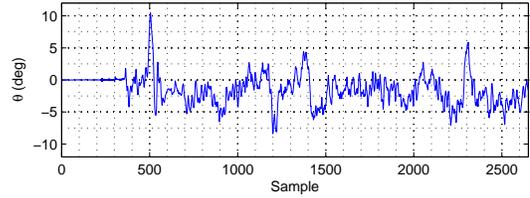
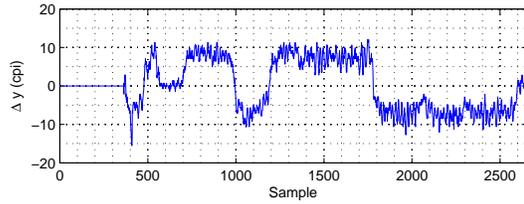
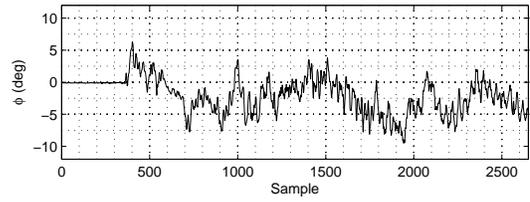
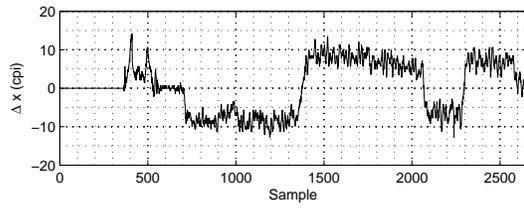
With the motors turned off and an approximate constant height of 1.5 meters, the path started on the marked red circle. The first shot aimed for no oscillations, meaning that the idea was to perform the route without any orientation (pitch, roll, yaw) changes.

At the end of the first test, a second test involved a similar linear path, but this time with oscillatory movements in pitch and roll, maintaining the heading.

Finally, with a fixed position, consecutive roll and pitch variations were forced, individually. Besides the data extracted from the sensor, the onboard IMU was also used.

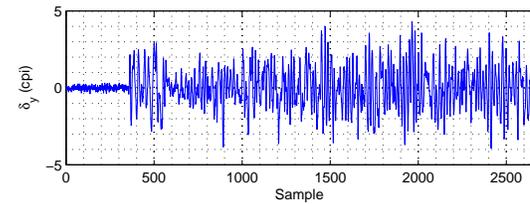
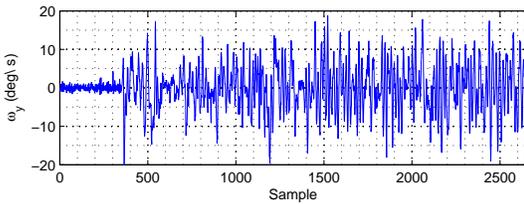
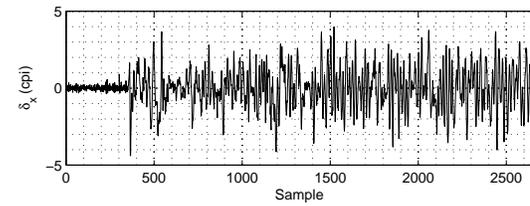
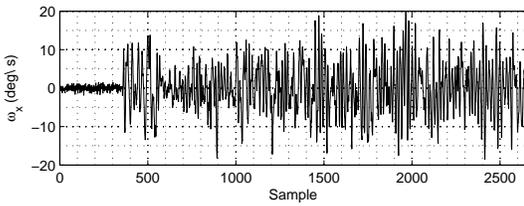
Since no moving platform was used and these tests were totally handmade, accuracy is not flawless. For instance, there were some height variations which could eventually lead to a certain bias in the outcome. Remember that height influences the sensitivity of the sensor, in the sense that the bigger the distance to the detected features, the less the optical flow values obtained (for constant velocity). However, the objective was to validate the sensor compensation.

Results presented in Fig.4.4 and Fig.4.5 are relative to the body frame of the quadrotor.



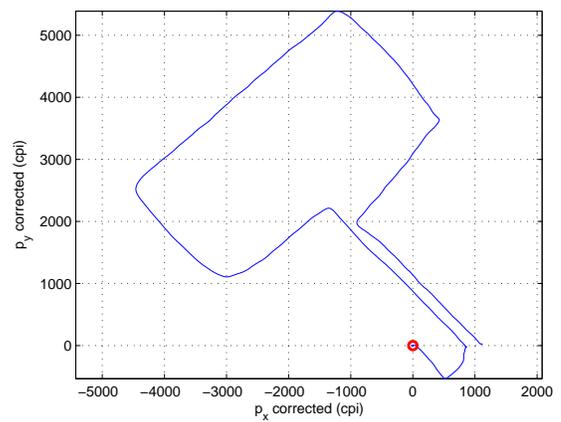
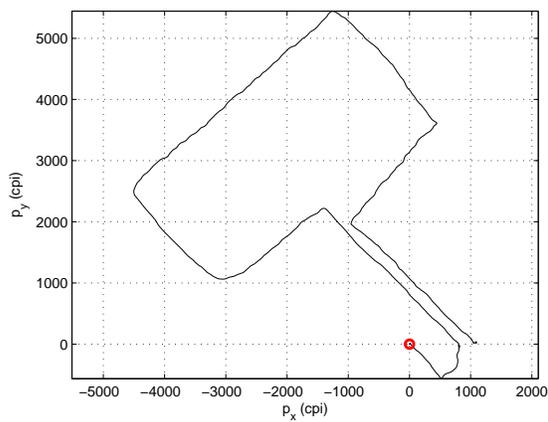
(a)

(b)



(c)

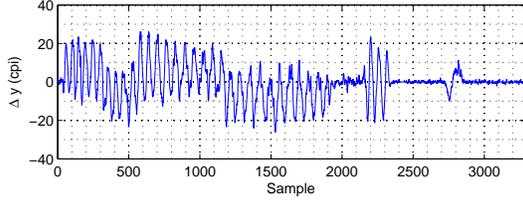
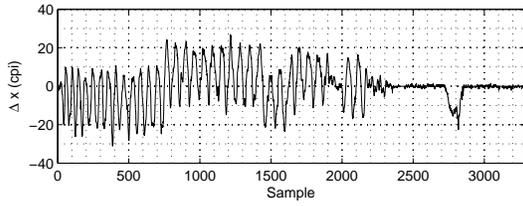
(d)



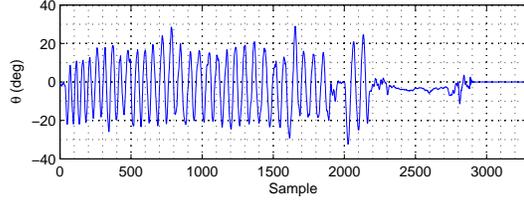
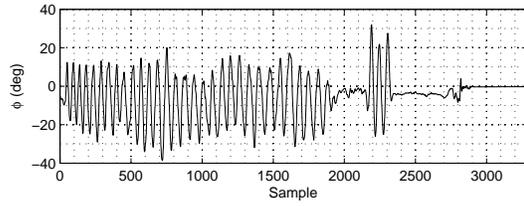
(e)

(f)

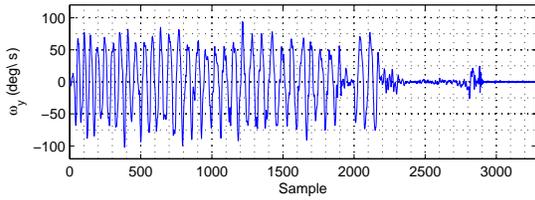
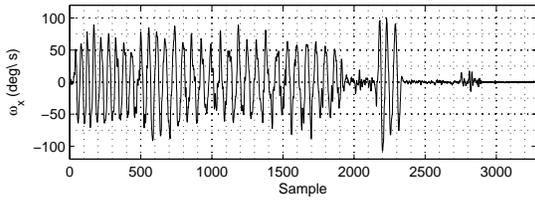
Figure 4.4: Linear path test. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position.



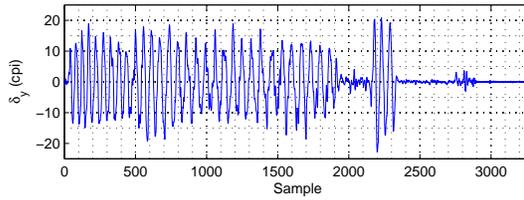
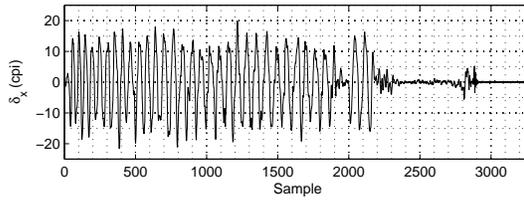
(a)



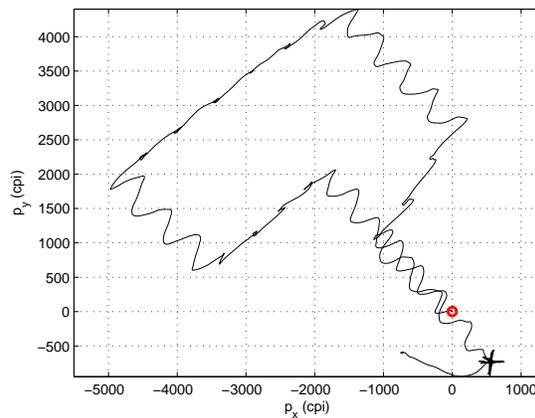
(b)



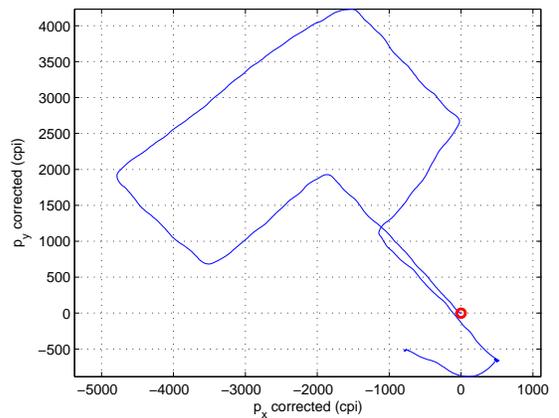
(c)



(d)



(e)



(f)

Figure 4.5: Oscillatory path test. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position.

By comparing the displacement error plots on both tests (Fig.4.4d and 4.5d), it is easy to evaluate the influence of rotations in the optical flow measurements, as the error in the second test is definitely bigger in both absolute and relative (to the measured displacement) values. An erroneous component is also present in the first test but this is due to the fact that the attitude of the quadrotor is not stable throughout the process, as manoeuvring was handmade. One should also add that the IMU data is a bit noisy, which leads to some compensation miscalculations. Nevertheless, it is still visible that the corrected position in the linear path test is better than the raw one.

The second test is definitely more aggressive, since the attitude is constantly changing. The compensation is really effective in this case, almost totally erasing any rotational effect in the optical flow measurements. This is reflected on both position plots shown (Fig.4.5e and 4.5f).

### 4.1.2 Simulator

A simulator of the quadrotor was developed by the DSOR team, presented in Fig.4.6.

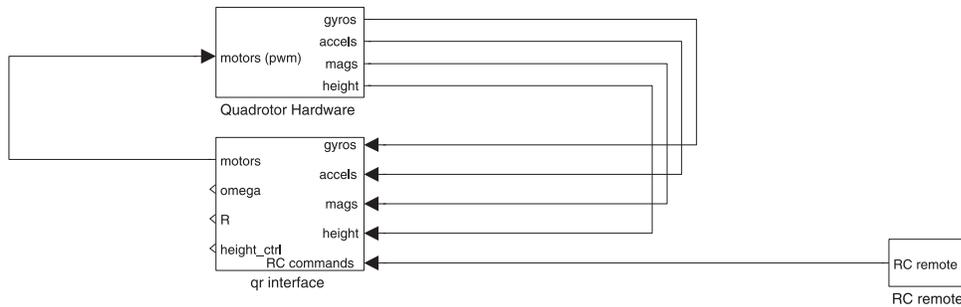


Figure 4.6: Simulink quadrotor model.

This model encompasses three main modules. The first is the quadrotor's model, which respects the dynamics described in the second chapter of this thesis. An RC command is also simulated and constitutes the second module. The last module is related to the control of the vehicle, running as an interface between the RC command and the quadrotor's model.

### 4.1.3 Optical Flow Controller

The next step is to define the process of incorporating the optical flow data in the system, in order to stabilize the quadrotor in hover position.

It is possible to generate optical flow sensor data based on the state of the quadrotor. Assuming that the SQUAL value is always enough to track enough visible features for optical flow calculation, the sensor comprises a translational component and a rotational component, given in the previous section by Eqs.4.2 and 4.4. Both are added and a unique estimate of the total optical flow is made.

The generated optical flow data can be compensated, according to the previous section, resorting to

the generated data from gyroscopes (recall that it measures, among other variables, the angular velocities  $\omega_x$  and  $\omega_y$ ), which includes added noise to simulate the real IMU device.

With the sensing side reliable, the final step is to develop a way of controlling the aerial platform based on the optical flow values registered on each sampling instant.

The concept of stabilizing the quadrotor resides on the control of the thrust vector. If the quadrotor moves left, for instance, the thrust vector should be such as to create a right horizontal force which opposes to the vehicle's velocity, while vertically balancing the gravitational force. This situation is depicted on Fig.4.7.

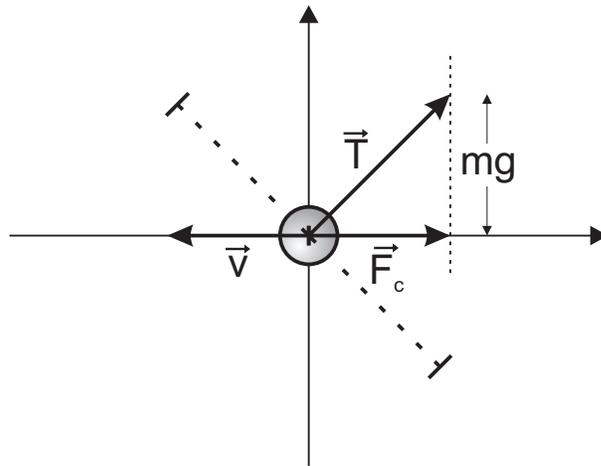


Figure 4.7: Method for controlling the quadrotor.

The desired control force  $\mathbf{F}_c$  in the body frame is then

$${}^B\mathbf{F}_c = -K_d {}^B\mathbf{v} \quad (4.6)$$

This force has two components,  $x$  and  $y$ . The velocity measured in the body frame along  $x$  and  $y$  is proportional to the measured optical flow (scaled by the sampling period and a unity conversion gain). Hence, the desired control force is proportional to the measured optical flow.

An attitude controller is already running in the quadrotor, developed by the DSOR team. One indicates the desired pitch, roll and yaw angles and the system reacts accordingly. There is also an altitude controller which stabilizes the quadrotor in respect to its height above ground.

It is possible to adapt the current system to the objective proposed of hover control. Imagine the quadrotor is moving at a constant positive speed  $v_x$ , relative to the body frame. By increasing the pitch angle, an horizontal backwards force in  $x$  is created, decelerating the vehicle - the same doesn't apply to  $v_y$  and roll. As  $z$ -axis points down, a positive roll tends to increase the  $y$ -component of  ${}^B\mathbf{F}_c$ . The idea is that instead of controlling forces through optical flow, the measured data can be fed into a loop as attitude deviations, as shown in Fig.4.8.

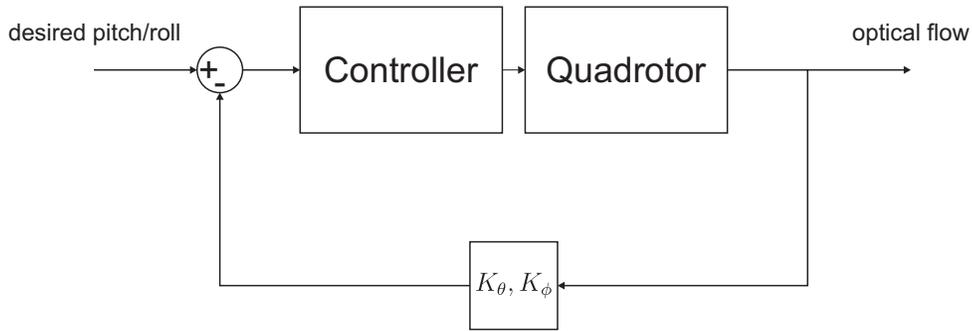


Figure 4.8: Implemented proportional controller.

The gains differ in sign. For the reasons described above,  $K_\theta$  is negative and  $K_\phi$  is positive.

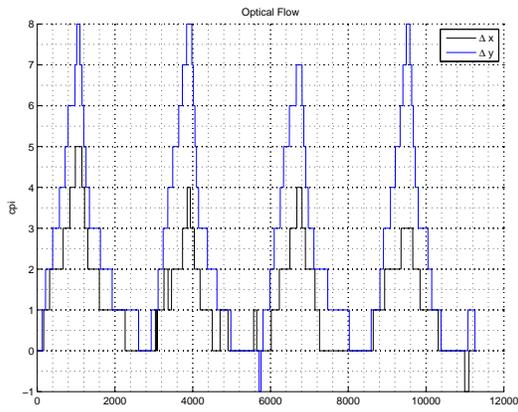
The process of using optical flow can then be described as:

- At time instant  $k$ , extract displacements in  $x$  and  $y$ , as well as the *SQUAL* value;
- Compensate rotational motions, by incorporating data from the IMU;
- Convert the translational optical flow in pitch/roll angles;
- Feed the RC Remote block with the desired angles.

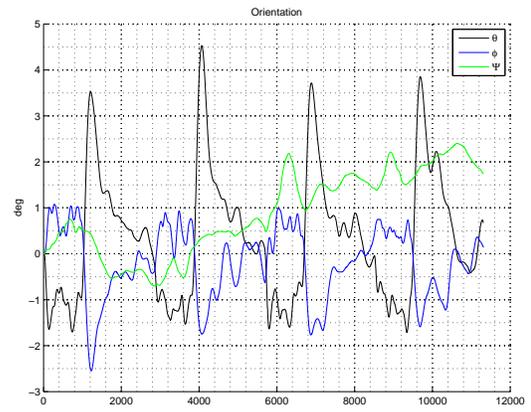
The proposed method works as a guidance process supported by the attitude controller running on the vehicle. Fig. 4.9 shows the simulation made using the described feedback process.

The controller was turned on/off four times, which explains the steps seen on the results. The idea was to let the quadrotor gain speed on both  $x$  and  $y$  (with different) rates and, periodically, let the controller actuate, to see how the system reacted.

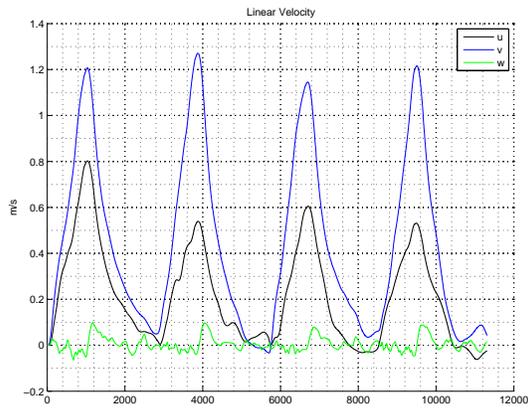
The influence of the gains is perfectly clear by inspecting the shown plots. When  $K_\theta, K_\phi = 5$  the system is very slow while stabilizing the position of the quadrotor. The increase of both gains improves the performance of the system, in the sense that it is quicker to react and stop the quadrotor's motion. However, if the gains are too high, one can be creating an oscillatory behaviour around the stabilized position. For  $K_\theta, K_\phi = 25$ , that is what happens. The effects of the gain on the system are justified by the change of the system poles. With minimum gains, the poles are distinct and real, with no imaginary components. There is a limit where the poles coincide and are real, and the system is critically damped. Finally, the oscillations are introduced by the imaginary components of poles, which appears after that limit is crossed.



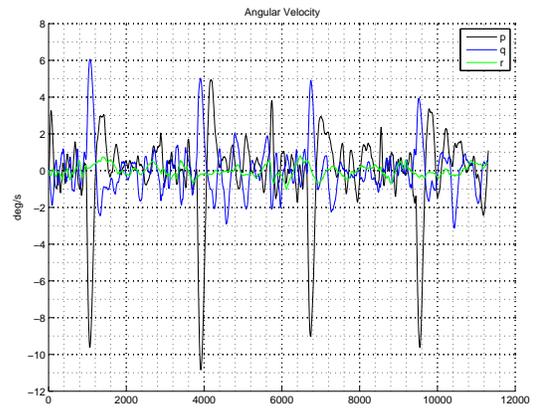
(a)



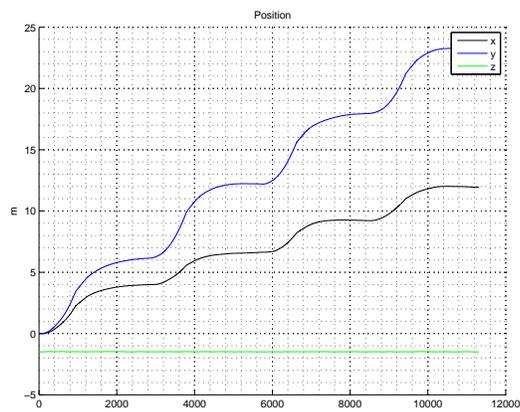
(b)



(c)

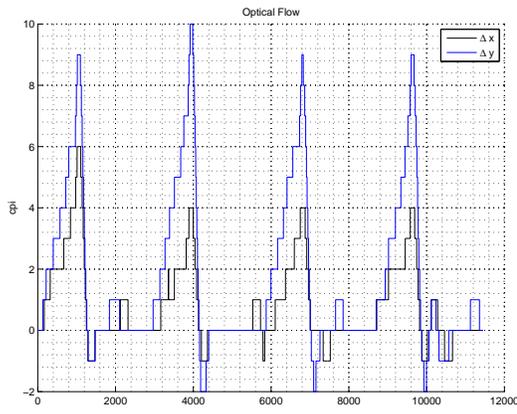


(d)

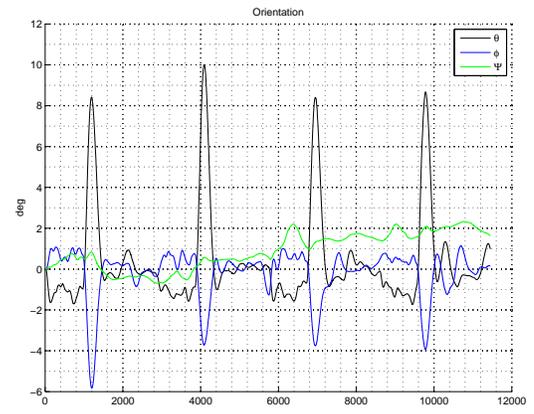


(e)

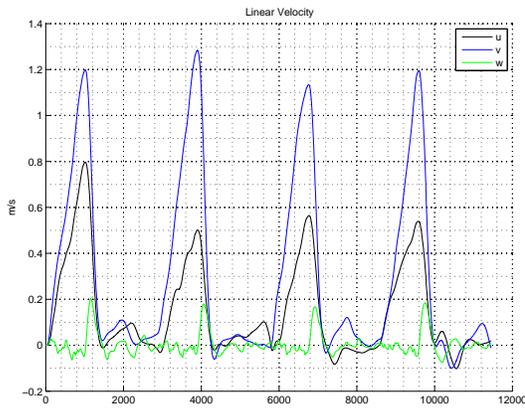
Figure 4.9: Simulation results for  $|K_\theta|, |K_\phi| = 5$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position.



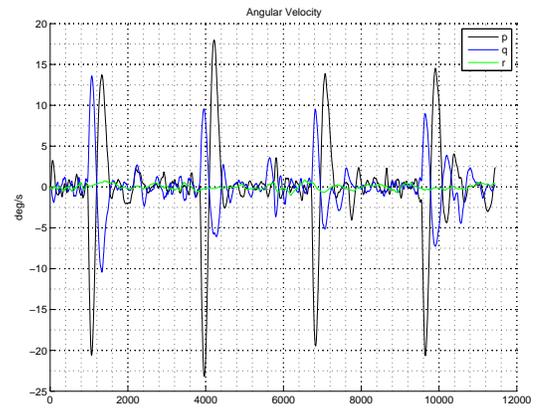
(a)



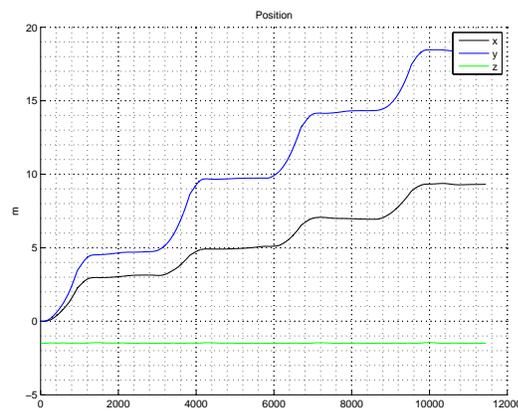
(b)



(c)

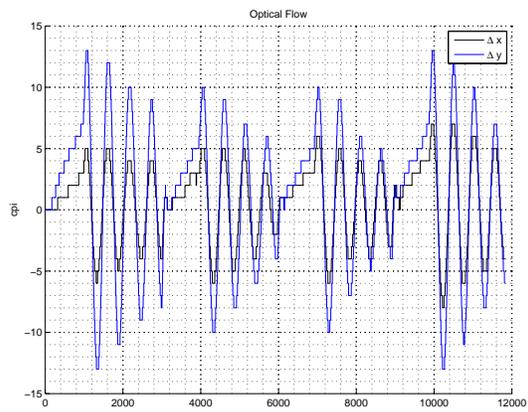


(d)

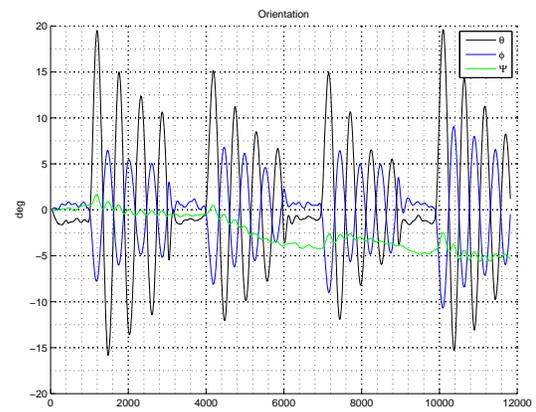


(e)

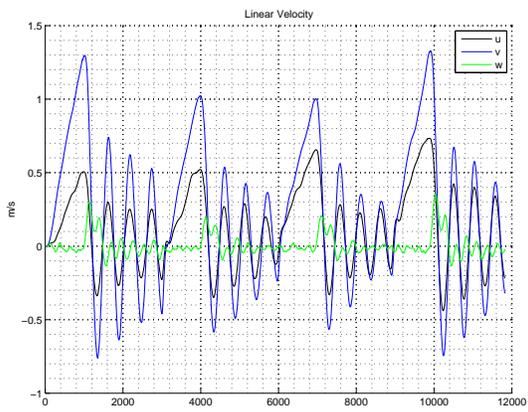
Figure 4.10: Simulation results for  $|K_\theta|, |K_\phi| = 12$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position.



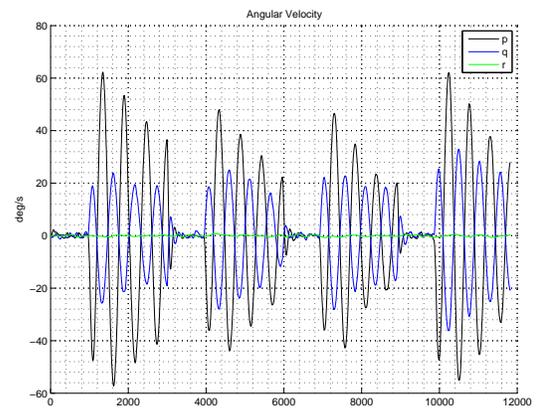
(a)



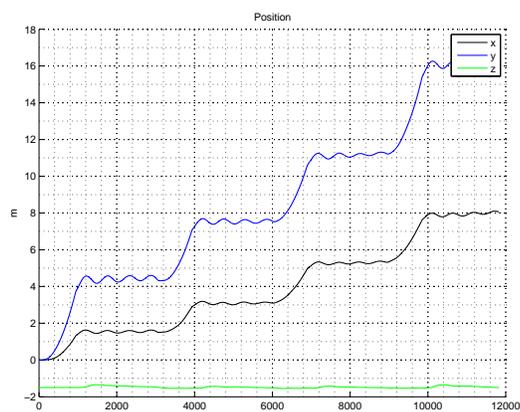
(b)



(c)



(d)



(e)

Figure 4.11: Simulation results for  $|K_\theta|, |K_\phi| = 25$ . (a) Optical flow simulated signal. (b) State euler angles. (c) Linear velocity. (d) Angular velocity. (e) Position.

Note one very important aspect about the optical flow measurements. As they are discrete, there is always a bias. Imagine the quadrotor is slowly drifting and the motion isn't detected by the sensor. This leads to a constant velocity towards some specific direction and not the hovering position. Although not obvious, due to the scale used in the presented results, the bias is present there. There is no obvious solution for this problem, unless another sensor is used, with higher sensitivity.

## 4.2 3D Model

The simulator lacked a way to visualize the simulation results in real time, besides the signal processing that could be done afterwards. For this reason, a 3D representation of the quadrotor was included in the Simulink model. Besides that, a Joystick Input block was also added, to engage the real scenarios. This block reads USB ports and automatically extracts all the information about the axis and buttons. It is only necessary to select the desired entries and use them at will. By connecting the correct axis to the remote RC block, the quadrotor can be freely guided by a human operator in the simulation, while visualizing the changes in a 3D representation. Interpreting the outcome of the process becomes easier to the human eye. Fig.4.12 shows the aforementioned 3D representation.

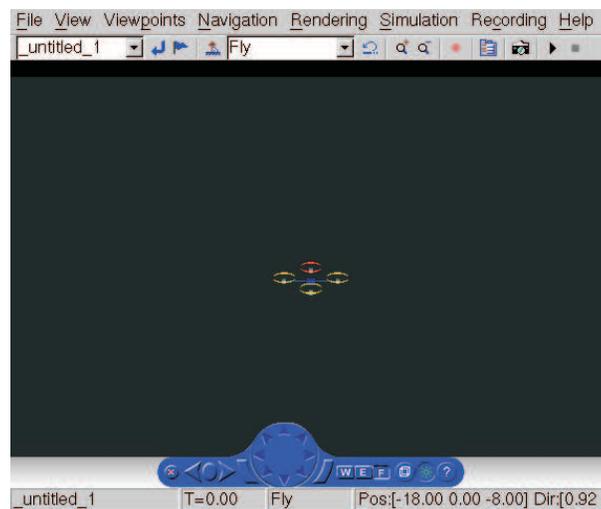


Figure 4.12: 3D Model added to the simulator.

## 4.3 Multiple Sensor Fusion

Considering the use of more than one sensor, the matter of fusing the data arises. As stated previously, these sensors also provide the quality measure - SQUAL - of the image for the user, which is directly related to the uncertainty of its information.

There are several approaches for this problem, one of each is to chose, from the set of working sensors, the one who presents the highest quality factor at a certain time instant. Alternatively, a weighted average can be computed. Considering a general case with  $N$  sensors, the displacement estimate in  $i$  direction at instant  $k$  is given by:

$$\widehat{\Delta}_i(k) = \frac{\sum_{j=0}^N SQUAL_j(k)\Delta_{i_j}(k)}{\sum_{j=0}^N SQUAL_j(k)} \quad (4.7)$$

## Chapter 5

# Conclusions and Future Work

The present document aimed at proposing a new way of controlling a quadrotor, using a lightweight cheap sensor, embedded in optical mice. The presented study in optics is good to dimensioning the lens for a certain application.

All alone, the sensor is insufficient, in the sense that it requires additional components to be able to successfully measure optical flow. On the one side, the illumination conditions must be sufficient enough to extract visible features. On the other side, the adapted lens is of utmost importance, since it determines the field of view of the sensor and, as such, it constrains the visible area. By diminishing the visible area, it increases magnification and the size of features in the image, losing the amount of incident light. The trade-off should be balanced, in order not to jeopardize the effectiveness of the sensor.

All in all, the potential of such sensors is amazing, in the sense that the whole image processing is available for free. The whole concept of camera plus computer can in fact be eliminated, when resorting to these devices. They are cheaper, lighter, quicker and less complex.

It is possible to control the vehicle in position using these sensors, as shown by the presented simulation results. Note that the control laws applied in this thesis were very basic, as was the guidance system. The main guideline of this thesis was to build a solid baseline in the use of these types of sensors in the quadrotor. Hence, the developed work is extensive and methodical concerning the ADNS-5050. By simply compensating the data extracted from the sensor, it can be directly used to detect translational motion and act accordingly.

### 5.1 Future Work

The defined objectives for this thesis weren't fully fulfilled.

As future goals, the controller should be ported to the real system and tested, to validate the simulation outcome. Not only the sensor should be used for hover control, but also for landing and obstacle avoidance. New structural dispositions of the sensor should be studied in order to support the control of such

manoeuvring. To avoid erroneous measurements due to error prone prototypes, new printed circuit boards (PCB) should be designed specially for the electronics in the presented system. This way, lens/sensor alignment errors are avoided.

The capabilities of the optical flow sensor should also be increased, by migrating the current system to the ADNS-3080, an optical flow sensor much more powerful than the ADNS-5050.

# Bibliography

- [1] Tommaso Bresciani. Modelling, identification and control of a quadrotor helicopter. Master's thesis, Lund University, 2008.
- [2] Stephen Griffiths. Remote terrain navigation for unmanned air vehicles. Master's thesis, Brigham Young University, 2006.
- [3] Geoffrey Barrows. Mixed-mode vlsi optical flow sensors for micro air vehicles. Master's thesis, University of Maryland, 1999.
- [4] Steven Cornelius Kriel. A comparison of control systems for the flight transition of vtol unmanned aerial vehicles. Master's thesis, University of Stellenbosch, March 2008.
- [5] Bruno Cardeira. Arquitecturas para navegação inercial/gps com aplicação a veículos autónomos. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, February 2009.
- [6] Randal W. Beard. Quadrotor dynamics and control. Technical report, Brigham Young University, 2008.
- [7] Geoffrey Barrows, Javaan Chahl, and M. Srinivasan. Biomimetic visual sensing and flight control. *The Aeronautical Journal*, 2003.
- [8] Antoine Beyeler, Jean-Christophe Zufferey, and Dario Floreano. optipilot: control of take-off and landing using optic flow. *Proceedings of EMMAV 2009*, 2009.
- [9] Simon Zingg, Davide Scaramuzza, Stephan Weiss, and Roland Siegwart. Mav navigation through indoor corridors using optical flow. *IEEE International Conference*, 2010.
- [10] Jean-Christophe Zufferey and Dario Floreano. Optic-flow-based steering and altitude control for ultra-light indoor aircraft. *Infoscience*, 2004.
- [11] Stefan Hrabar and Gaurav Sukhatme. A comparison of two camera configurations for optic-flow based navigation of a uav through urban canyons. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

- [12] Bruno Herisse, Sophie Oustrieres, Tarek Hamel, Robert Mahony, and Francois-Xavier Russotto. A general optical flow based terrain-following strategy for a vtol uav using multiple views. *IEEE International Conference on Robotics and Automation*, 2010.
- [13] Ronald Chan, Aamir Mulla, and Karl Stol. Characterisation of low-cost optical flow sensors. *IEEE International Conference on Robotics and Automation*, 2010.
- [14] Nicolas Francheschini, Franck Ruffier, Julien Serres, and Stéphane Viollet. *Optic flow based visual guidance: from flying insects to miniature aerial vehicles*, pages 747–770. InTech, 2009.
- [15] Euler angles. [http://en.wikipedia.org/wiki/Euler\\_angles](http://en.wikipedia.org/wiki/Euler_angles), July 2011.
- [16] Rotation (mathematics). [http://en.wikipedia.org/wiki/Rotation\\_\(mathematics\)](http://en.wikipedia.org/wiki/Rotation_(mathematics)), July 2011.
- [17] Farrel J. *Integrated Aircraft Navigation*. Academic Press Inc, New York, 1976.
- [18] Jonghyuk Kim and Galen Brambley. Dual optic-flow integrated navigation for small-scale flying robots. *ACT 0200*, 2007.
- [19] John E. Bortz. A new mathematical formulation for strapdown inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 1970.
- [20] Avago Technologies. *ADNS-5050 - Optical Mouse Sensor*.
- [21] ATMEL. *8-bit Microcontroller with 32K/64K/128K Bytes of ISP Flash and CAN Controller - AT90CAN32/AT90CAN64/AT90CAN128*.
- [22] Dieter Flubacher. Characterisation of an optical flow sensor for off-road robot application. Master's thesis, Swiss Federal Institute of Technology Zurich, 2008.
- [23] Xiaojing Song, Lakmal D. Seneviratne, and Kaspar Althoefer. A kalman filter-integrated optical flow method for velocity sensing of mobile robots. *IEEE/ASME Transactions on Mechatronics*, 2011.
- [24] Daisuke Sekimori and Fumio Miyazaki. Self-localization for indoor mobile robots based on optical mouse sensor values and simple global camera information. *IEEE International Conference on Robotics and Biomimetics*, 2005.
- [25] Sungbok Kim and Sanghyup Lee. Robust mobile robot velocity estimation using redundant number of optical mice. *Proceedings of the 2008 IEEE International Conference on Information and Automation*, 2008.
- [26] Michael Dille, Ben Grocholsky, and Sanjiv Singh. Outdoor downward-facing optical flow odometry with commodity sensors. *Proceedings of Field & Service Robotics*, 2009.

- [27] Andrea Bonarini, Matteo Matteucci, and Marcello Restelli. Automatic error detection and reduction for an odometric sensor based on two optical mice. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005.
- [28] Paul Pounds, Roberto Mahony, and Peter Corke. Modelling and control of a quad-rotor robot. *CSIRO ICT Centre*, 2005.
- [29] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.



# Appendices



# Appendix A

## Communicating with ADNS-5050

As mentioned earlier in this thesis, the solution for communicating with the sensor was handled by the developed drivers and is usually called bit banging. Here, all the lines are directly controlled by the micro-controller and the tabled timings for operations have to be respected. For this, the manual provided by Avago was very important [20], not to mention the microcontroller's manual itself [21].

Just to provide a short brief catch of the main operations, this section presents the two main operations over the sensor (read and write) and the timings required for each of them.

Firstly, the write operation consists of data flowing to the sensor. The microcontroller initiates the operation by sending an 8-bit address, with the most significant bit (MSB) as binary "1", to indicate the direction of data. The second byte contains the data to be written on the sensor. This operation is depicted on Fig.A.1.

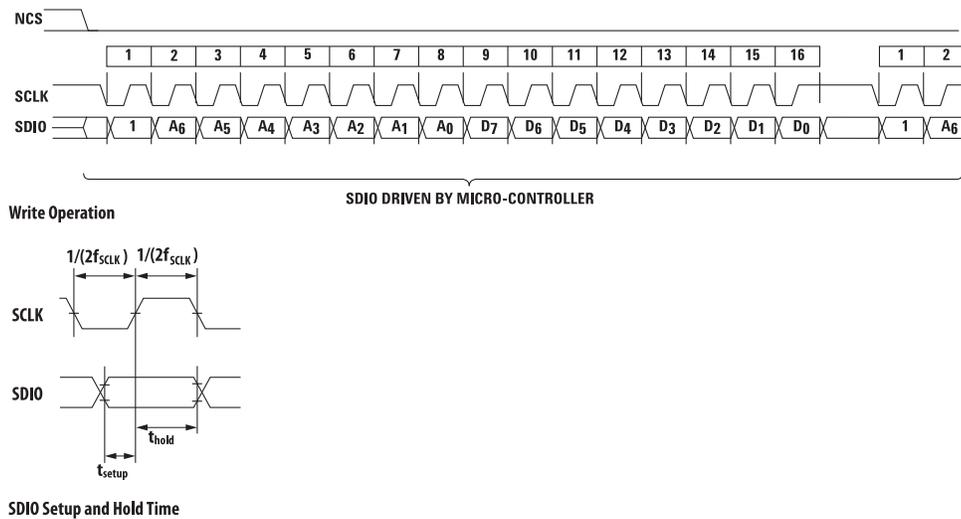


Figure A.1: Write operation.

This can be considered the most simple operation, as the communication channel is always direct to

the same side. Another important operation is the read operation, where data flows from the sensor to the microcontroller. Once again, the latter controls the communication, which is consisted of two bytes. The first is written on the line by the microcontroller and contains the address (the MSB is set to binary "0", to indicate data direction). The second byte is the triggered response of the sensor, driven over SDIO. The direction is changed in the middle of communication, unlike in a write operation.

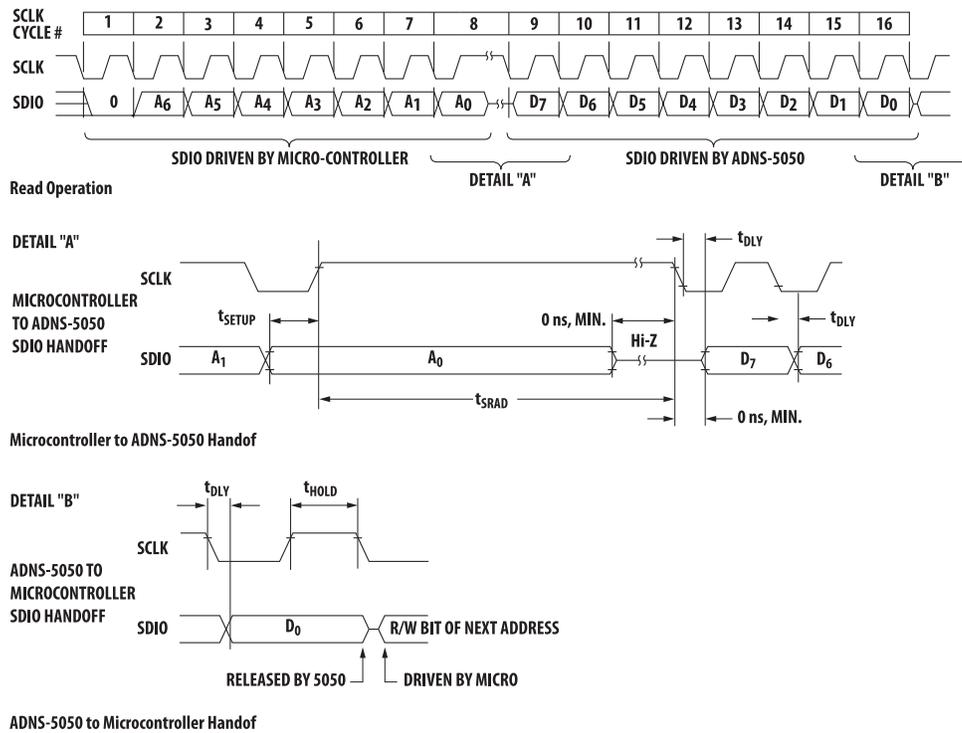


Figure A.2: Read operation.

There are also some timing restrictions to consecutive operations. These are summed up in Tab. A.1 and temporal diagrams are shown in Fig.A.3.

Symbol	Description	Minimum	Maximum
$f_{SCLK}$	Serial port clock frequency	-	3MHz
$t_{r-SDIO}$	SDIO rise time	-	300ms
$t_{f-SDIO}$	SDIO fall time	-	300ms
$t_{DLY-SDIO}$	SDIO delay after SCLK	-	120ns
$t_{hold-SDIO}$	SDIO hold time	100ns	-
$t_{setup-SDIO}$	SDIO hold time	200ns	-
$t_{SWW}$	Time between write commands	30ns	-
$t_{SWR}$	Time between write and read commands	20ns	-
$t_{SRW}/t_{SRR}$	Time between write and subsequent commands	500ns	-
$t_{SRAD}$	Time between read address and data delay	$4\mu s$	$10\mu s$
$t_{NCS-SCLK}$	NCS to SCLK active	120ns	-
$t_{SCLK-NCS}$	SCLK to NCS inactive (for read operation)	120ns	-
$t_{SCLK-NCS}$	SCLK to NCS inactive (for write operation)	20ns	-
$t_{NCS-SCLK}$	NCS to SCLK high-Z inactive	-	500 ns

Table A.1: Timing restrictions of the communication protocol with ADNS-5050.

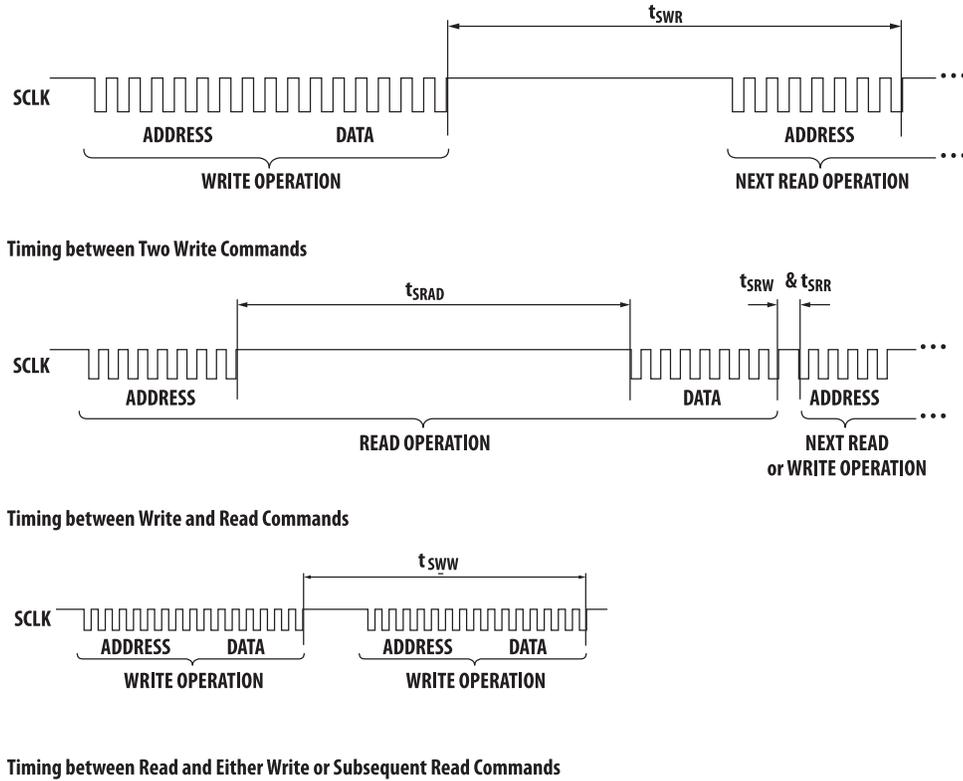


Figure A.3: Operation's timings.



## Appendix B

# Compensation of rotational components

Another compensation test is presented in this section. The diagram of the experiment can be seen on FigB.1 and real results on Fig.B.2.

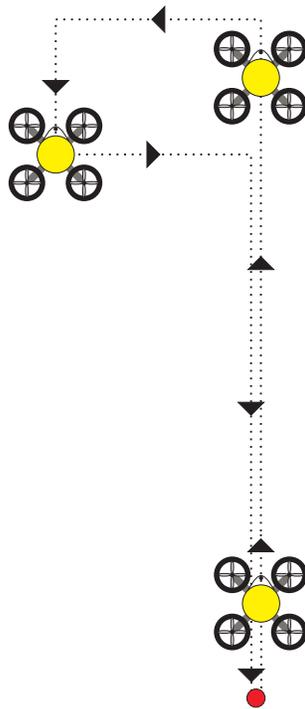
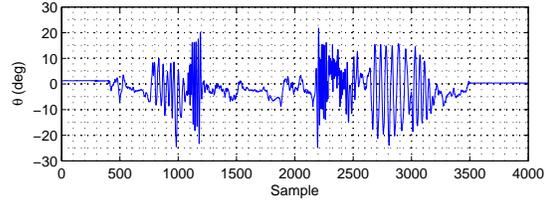
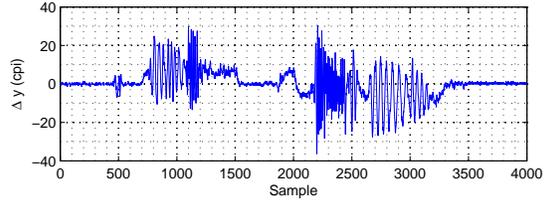
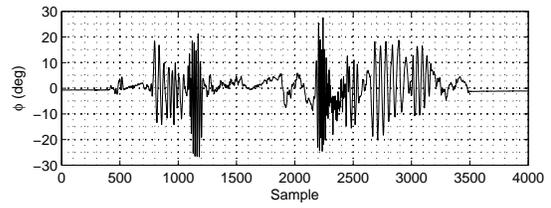
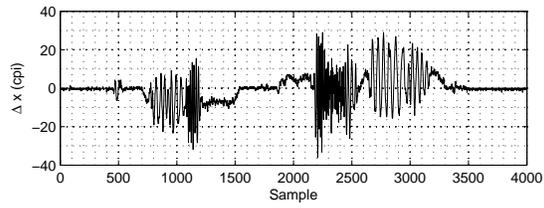
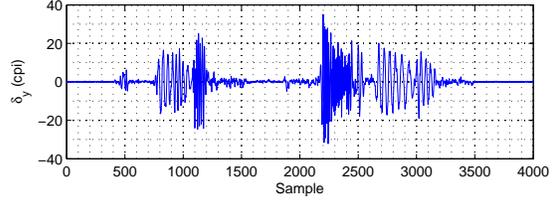
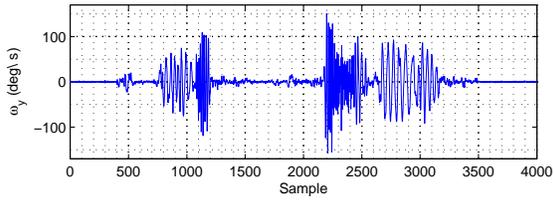
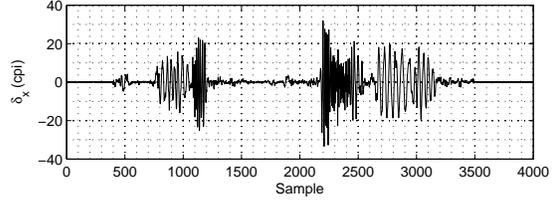
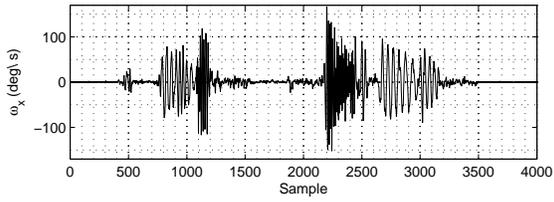


Figure B.1: Optical flow compensation test.



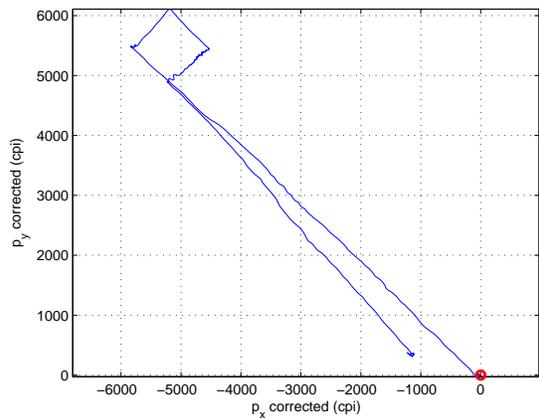
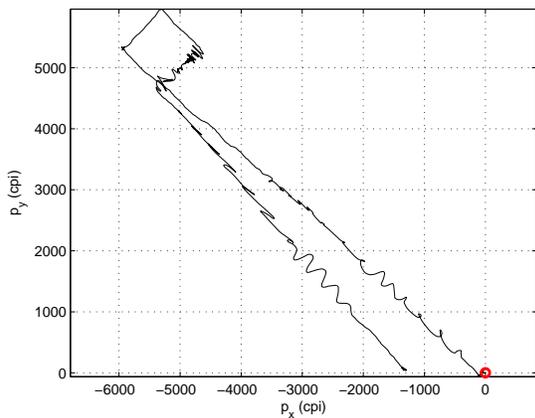
(a)

(b)



(c)

(d)



(e)

(f)

Figure B.2: Test for the compensation of rotational motions. (a) Displacements in x and y. (b) Euler angles (IMU). (c) Angular velocities in x and y (IMU). (d) Displacement error. (e) Raw position. (f) Compensated position.

This test was useful to show the influence of the optical flow correction in the raw data. Once again, while performing the indicated path, forced oscillatory rotational movements were applied to the quadrotor. As in the test shown in Section 4.1.1, the presented algorithm is capable of erasing most of the rotational components of the measured optical flow. Notice the oscillations that weren't totally removed on one of the edges of the square. On that particular side, the induced rotational movements were very sudden and quick, unlike the quadrotor's dynamic when flying. There is also a bias, evidenced by the difference between the start and end position (they were physically the same). One possible explanation besides the sensor's noise is the fact that yaw wasn't compensated. The whole process assumed the yaw rate was null.



# Appendix C

## Root Locus

In order to theoretically infer something about the developed controller, a root locus analysis is present in this appendix.

Considering the absence of position feedback and the use optical flow to control the vehicle's velocity, one can approximate the vehicle's system to the following:

$$\begin{aligned}\dot{v}_x &= \frac{1}{m}f_x \\ \dot{v}_y &= \frac{1}{m}f_y\end{aligned}\tag{C.1}$$

where the expressions for the controller are

$$\begin{aligned}f_x &= -K_\theta(v_x - v_x^{REF}) \\ f_y &= -K_\phi(v_y - v_y^{REF})\end{aligned}\tag{C.2}$$

On the frequency domain, this is equivalent to

$$\begin{aligned}V_x(s) &= G(s)F_x(s), & G(s) &= \frac{1}{ms} \\ F_x(s) &= -K_\theta(V_x(s) - V_x^{REF}(s))\end{aligned}\tag{C.3}$$

and

$$\begin{aligned}V_y(s) &= G(s)F_y(s), & G(s) &= \frac{1}{ms} \\ F_y(s) &= -K_\phi(V_y(s) - V_y^{REF}(s))\end{aligned}\tag{C.4}$$

Finally, the transfer function for both feedbacks is

$$\begin{aligned}H_x(s) &= \frac{V_x}{V_x^{REF}} = \frac{G(s)K_\theta}{1+G(s)K_\theta} \\ H_y(s) &= \frac{V_y}{V_y^{REF}} = \frac{G(s)K_\phi}{1+G(s)K_\phi}\end{aligned}\tag{C.5}$$

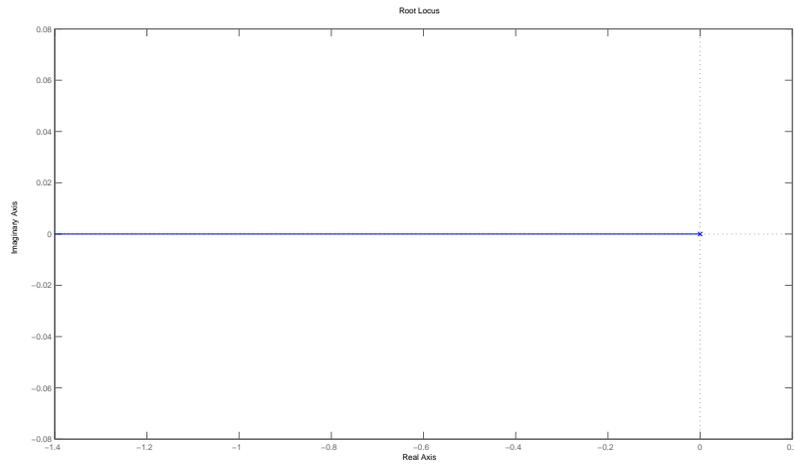


Figure C.1: Root locus of one of the transfer functions.

The root locus is presented in Fig.C.1.

The results are not conclusive, due to the simplicity of the method used and the adopted approximations. The poles indicate the stability of the system, independently from the value of the gains, nevertheless.

The gains should be adjusted using the values withdrawn from the dimensioning made by simulation.

# Appendix D

## 12 mm Lens vs 16 mm Lens

In order to compare the effectiveness of both lenses on the quadrotor with ADNS-5050, a test was performed using two sensors with two different lenses (12mm and 16mm). A sheet of newspaper was placed on the floor and the quadrotor was moved at a height of approximately 1.5 meters, front and back, on top of the sheet.

The next figure shows the quality measure of both sensors.

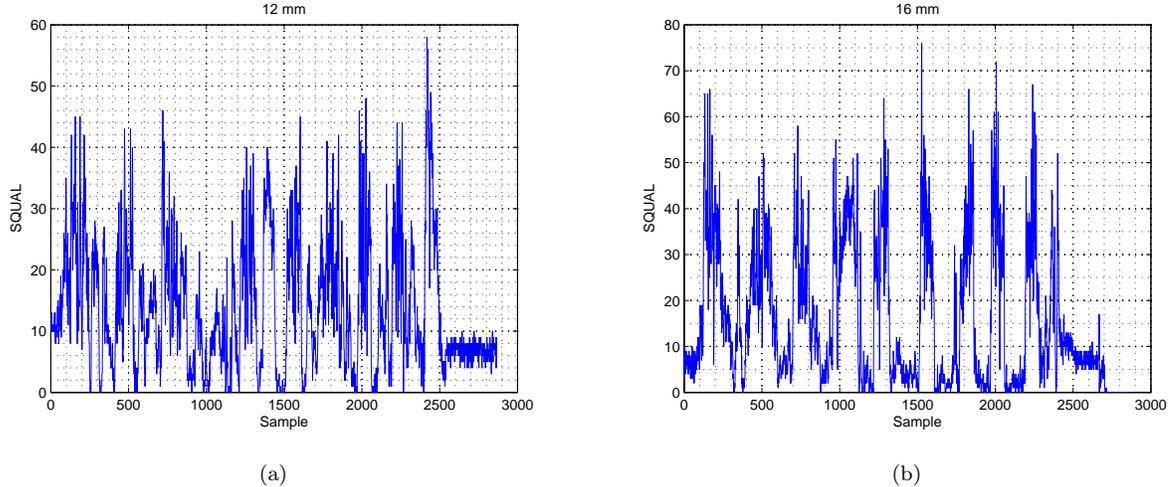


Figure D.1: Test for the comparing two different lenses and their effectiveness. (a) Using a lens with  $f = 12$  mm. (b) Using a lens with  $f = 16$  mm.

As expected, the signal of quality oscillates between a fairly reasonable amount and poor quality, due to the fact that the sensors were moving in and out of the sheet. When the sensor comprised the sheet, the quality was good. Outside the sheet, the dark brown floor represented a low amount of distinguishable features.

These results are enough to confirm that the 16mm lens is better for the proposed height, which means

that it will be a better choice for controlling the quadrotor indoors. Although the lighting conditions are better for that lens (remember, as the focal distance decreases, the field of view increases and so does the amount of light), the 16mm lens still accomplishes better results for quality rates. As such, this should be the adopted set-up for the quadrotor.