Luis Rodolfo García Carrillo
Alejandro Enrique Dzul López
Rogelio Lozano
Claude Pégard

Quad Rotorcraft Control

Vision-Based Hovering and Navigation

AIC

Springer

Advances in Industrial Control

Luis Rodolfo García Carrillo ·
Alejandro Enrique Dzul López ·
Rogelio Lozano · Claude Pégard

Quad Rotorcraft Control

Vision-Based Hovering and Navigation

Luis Rodolfo García Carrillo
HEUDIASYC UMR 6599, Centre
  de Recherches de Royalieu
Université de Technologie de Compiègne
Compiègne cedex, France

Alejandro Enrique Dzul López
División de Estudios de Posgrado
Instituto Tecnológico de la Laguna
Torreón, Mexico

Rogelio Lozano
UMR-CNRS 6599, Centre de Recherche
  de Royalieu
Université de Technologie de Compiègne
Compiègne, France

Claude Pégard
Laboratoire MIS EA 4290
Université de Picardie Jules Verne
Amiens, France

To our families

Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies..., new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

Control engineers and academics have made significant contributions to the control of Autonomous Underwater Vehicles (AUVs). Because of their critical deployment in the offshore industry, this continues to be a strong and developing technological field. Turning now to Unmanned Aerial Vehicles (UAVs), there is a rather different history of development and deployment. Military applications have been a significant driving force for this technological field, and civilian applications, despite their potential importance, have been muted by comparison. One factor in this has been that unlike the ocean deeps, civilian airspace is populated by large aircraft carrying significant numbers of people whose safety cannot be jeopardized by small, possibly unpredictable, UAVs. Consequently licensing UAVs to operate in civilian airspace on a routine basis remains an issue to be resolved.

Nonetheless, control engineers and academics have been contributing to the development of control systems for UAVs and more and more of this work is now appearing in monograph form. The *Advances in Industrial Control* monograph series has always promoted reports of current applications as well as research work that shows a potential for future application. Consequently within the series there is a small but growing set of monographs that report on the control developments for UAVs, including:

2011 G. Cai, B.M. Chen and T.H. Lee, *Unmanned Rotorcraft Systems*, ISBN 978-0-85729-634-4;

2009 G.J.J. Ducard, *Fault-tolerant Flight Control and Guidance Systems*, ISBN 978-1-84882-560-4;

2005 P. Castillo, R. Lozano and A.E. Dzul, *Modelling and Control of Mini-Flying Machines*, ISBN 978-1-85233-957-9; and

2003 A. Isidori, L. Marconi and A. Serrani, *Robust Autonomous Guidance*, ISBN 978-1-85233-695-0.

The series also expects to feature some future monographs in this field, including:

A. Abdessameud and A. Tayebi, *Motion Coordination for Aerial Vehicles* (in preparation)

Quad Rotorcraft Control: Vision-Based Hovering and Navigation by Luis R. García Carrillo, Alejandro E. Dzul López, Rogelio Lozano, and Claude Pégard now adds to this valuable set of monographs in *Advances in Industrial Control*. At this point, some clarification is perhaps needed; rotorcraft take many different guises, some have the well-known "standard" helicopter rotor configuration whilst there are other configurations such as the aerial vehicle activated by four (quad) sets of rotors. It is a quad rotorcraft that is the focus here. Another key feature of this monograph is the use of vision to control the hovering and navigation functions of the vehicle. The editors of the series have long sought some monographs that report on how vision can be integrated into the control system. Despite a long search, this is the first monograph in the series that treats this particular control-technological development.

In the monograph the authors use the first three chapters to create the framework for the research to be reported. This includes an introduction that contains a valuable historical and state-of-the-art review (Chap. 1), a progression of the system modelling aspects (Chap. 2) and a description of the experimental platform and the existing control loops (Chap. 3).

From there the authors present their own research: non-vision-based techniques to improve hover performance (Chap. 4), the hardware and implementations for the camera system and image analysis of the vision system (Chap. 5), and vision-based control (Chaps. 6 and 7). Videos of some of the experimental work can be found on YouTube. Finally, the authors present some directions for future research and developments in a concluding chapter (Chap. 8).

The monograph presents research that complements existing and future series monographs in on unmanned aerial vehicles. The Editors are very pleased to welcome this monograph into *Advances in Industrial Control* and are particularly gratified at last to be able to add to the series a work that explores the technology of vision-based control.

Industrial Control Centre,   M.J. Grimble
Glasgow, Scotland, UK   M.A. Johnson

Foreword

Unmanned Aerial Systems (UAS) have experienced in recent years an important growth both in research activities and in the industrial development of platforms to be used in applications.

Today, Unmanned Aerial Vehicles (UAVs) range from more than ten tonnes weight and tens of meters wingspan, to micro and even nano UAVs with grams of weight and few centimeters, or even millimeters, wing span.

The systems with multiple rotors have attracted significant attention. The most popular are the quad-rotors but others with six and eight rotors have been developed. These multiple rotor systems are able to hover and have good maneuvering capabilities. From the mechanical point of view they can be considered simpler than helicopters because they do not have the swash-plate and do not need to eliminate the gyroscopic torques created by the spinning motors. Moreover, the rotors develop less energy than the equivalent main rotor of the helicopters and then are safer than helicopters and can be protected to fly in close proximity to people. Thus, they are familiar in many research laboratories and also are becoming attractive for many applications that do not require significant payloads.

The group led by Prof. Rogelio Lozano at the Université de Compiègne is playing a significant role in the research and development activities of mini UAV systems, and particularly in quad-rotor technologies and control systems. This book reports doctoral research by the first author along with additional contributions and improvements.

Modeling is needed for the development of guidance, navigation and control systems, as is also relevant for many applications. The second chapter of this book is devoted to the modeling of quad-rotors. A general overview of the quad-rotor mini-rotorcraft and its operation principle is given. Next, the quad-rotor modeling is addressed using Euler–Lagrange and Newton–Euler methods. The Lagrange equations obtained from Newton's equations are also shown. Finally, the Newton–Euler modeling for an "X-Flyer" quad-rotor configuration is presented.

The third chapter presents an experimental quad-rotor system. This consists of the vehicle and a supervisory ground station where image processing and control algorithms are executed. General details concerning the most common sensing tech-

nologies available on UAVs are given, as well as the architecture of these kinds of experimental platform. The quad-rotors conceived at Compiègne are described in this chapter. The hierarchical control strategy allows stabilizing the quad-rotor during autonomous flights. The performance of the vehicles is validated in real-time experiments.

Chapter 4 introduces an embedded control system for improving the attitude stabilization of a quad-rotor. The proposed control strategy uses low cost components and includes an extra control loop based on motor armature current feedback. The control strategy presented here is robust with respect to external disturbances. Experimental results performed indoors have shown that the additional control loop significantly improves the performance of the quad-rotor attitude stability.

Most navigation and guidance strategies of UAS are based on satellite positioning systems. However, the reliability of these systems is a critical issue for safety. Satellite occlusions, multi-paths and other radio communication problems led to the need of supplement or substitute the satellite positioning systems by other alternative, which is mandatory in indoor and other GPS denied environments. In this context the application of image processing in guidance, navigation and control of UAS is a main topic.

This book also covers in Chap. 5 image sensors for state estimation. Basic background concerning computer vision is first given. The pinhole camera model, as well as the camera calibration procedure is shown. Stereo imaging, together with a method for stereo calibration and rectification are also presented. The concept of optical flow and a method for its computation are detailed. The chapter analyses both the onboard implementation on embedded systems and in the ground station. Relevant issues that should be considered when implementing an imaging system onboard a quad-rotor UAV are discussed. Finally, the hardware and software development of both a monocular and a stereo imaging system are presented.

Chapter 6 introduces two different vision-based control strategies for stabilizing a quad-rotor during flight. The first strategy is based on a homography estimation technique and an optical flow computation. Using this approach, a comparison of three control methods is addressed with the purpose of identifying the most effective approach for stabilizing the vehicle when using visual feedback. In the second strategy, the vision system is implemented for altitude control stabilizing the 3-dimensional position and regulating the velocity of the vehicle using optical flow. Real-time experiments of autonomous hover and navigation are executed for validating the effectiveness of the two vision-based control strategies.

Chapter 7 is devoted to combining sensing systems for the quad-rotor. Particularly, stereo imaging, inertial and altitude sensors are combined. The objective is to enable the vehicle to autonomously perform take-off, relative positioning, navigation and landing. A real-time comparison study between a Luenberger observer, a Kalman filter and a complementary filter is also addressed, with the purpose of identifying the most effective approach for combining the different sensing technologies.

In summary, the book combines modeling, control and computer vision and their implementation for quad-rotor guidance, navigation and control. It presents signif-

icant work and comparative results in an important topic relevant for quad-rotor system applications indoor and in other GPS-denied environments.

This book will contribute to the development of aerial robotics and unmanned aerial systems. It will be very useful not only for research and development activities but also for Master and other academic courses as well as for engineers interested in the development and application of quad-rotor systems.

Seville, Spain   Anibal Ollero

Preface

Unmanned Aerial Vehicles (UAVs) are developed today to perform more demanding tasks. Both civil and military applications require autonomous vehicles with increased performance and capabilities. Modeling and nonlinear controller syntheses have been essential tools for stabilizing the orientation of helicopters using measurements from an IMU (Inertial Measurement Unit). Such techniques have been used for stabilizing different configurations including the classical helicopter, quadrotors as well as other types of rotorcrafts. In order to stabilize a helicopter at a desired hovering position we require a position measurement system. GPS (Global Positioning System) is one of the most common position sensors. However, GPS position measurement error can be significant due to obstacles (buildings, weather conditions, etc.). In GPS denied areas Computer Vision can be used as an alternative position measurement system.

This book is devoted to study how a computer vision system can be used onboard to estimate the translational velocity of the aircraft and the position of the helicopter with respect to landmarks in the environment. Horizontal velocity of the rotorcraft is achieved by using the optical flow measurement in the feedback control law. The tasks of hovering at a desired point as well as velocity regulation are accomplished by using the position and velocity measurements with respect to landmarks in the environments. Stereo vision is also studied with the purpose of enabling the aerial vehicle to localize itself in unstructured indoors environments. The different approaches proposed have been tested in experimental platforms which are illustrated in this book.

Santa Barbara, California, USA — Luis Rodolfo García Carrillo

Torreón, Coahuila, México — Alejandro Enrique Dzul López

Compiègne, France — Rogelio Lozano

Amiens, France — Claude Pégard

Acknowledgements

Santa Barbara, California, USA	Luis Rodolfo García Carrillo
Torreón, Coahuila, México	Alejandro Enrique Dzul López
Compiègne, France	Rogelio Lozano
Amiens, France	Claude Pégard

Contents

Chapter 1
Introduction

In recent years, researchers and engineers from diverse areas have worked intensively to develop effective flying machines capable of performing missions with minimum or no human intervention. This kind of vehicle is commonly known as Unmanned Aerial Vehicle (UAV). Control theory, computer vision, mechanics, aerodynamics, automation and embedded electronics are some of the fields related to the development of these systems.

This chapter presents first a brief summary of the UAVs in a historical way, followed by an explanation of the more common applications and classification of these kinds of vehicle. This previous sections will allow the reader to locate in its right context the quad-rotor platform used for the present research. The chapter continues with a discussion concerning the quad-rotor platform state of the art, as well as with some issues and problems still considered a challenge for engineers and researchers working with this kind of vehicle. Finally, a brief description of the content of this book is outlined.

1.1 Unmanned Aerial Vehicles

An Unmanned Aerial Vehicle (UAV) is defined as a crewless aircraft, which can be driven by a pilot at a ground control station or can fly autonomously based on pre-programmed flight plans or more complex dynamic automation systems. UAVs can carry a variety of payloads according to their requirements. Furthermore, they can be expendable or recoverable.

An interesting characteristic of UAVs is their capacity to communicate valuable information, such as temperature, images, or video of its environment. Furthermore, these systems can provide its primary state data, concerning position, speed, heading and altitude, remaining fuel or energy, temperatures of components, e.g. engines or electronics. UAVs are mainly appreciated since they can accomplish a large group of civilian and military applications without putting human lives at stake. Also, since they are free of aircrew, these vehicles can be designed to be smaller, which favors an easier storage and transportation.

During early stages of the development of UAVs, these systems were called drones due to their autonomy restrictions. Those drones needed a constant supervision from a pilot that controlled their flight via radio commands. Nowadays, scientific progress within the embedded control and guidance systems of the UAVs allow them to autonomously perform tasks such as take-off and landing, flight stabilization and point-to-point navigation.

1.1.1 Brief History

UAVs were first seen as a possibility to counteract offensive and defensive actions of opponents. Subsequently, UAVs capabilities were expanded, as they were touched by the advancements in technology, mainly due to electronics and logistics of guidance and control [9]. The first UAVs were developed as long-range armament and are considered as the forerunners of cruise missiles, see Fig. 1.1. In 1917, the US Navy presented the "Aerial Torpedo", a pilotless biplane bomber made of wood, weighing 270 kg and powered by a 40 horsepower engine from Ford. The characteristics of this vehicle can be listed as: a gyrostabilizer to keep the aircraft level, an automatic steering gyro to keep the aircraft on a preset heading, a barometer to indicate cruise altitude, causing the aircraft to level off, an engine revolution counter to determine when the aircraft should cut power and dive into its target. Also, a wind-driven electrical generator was used to provide power for the gyro motors and the servomotors that moved the aerial torpedo's flight control surfaces. The "Kettering Bug" was a similar but lighter biplane presented by the US Army in 1918. Similar to the "Aerial Torpedo", it was designed to carry an explosive payload of 82 kg. The British Army's "Aerial target", from 1914, was a radio-controlled pilotless monoplane, built with the purpose of proving the effectiveness of using radio signals to guide a flying bomb to its target. The developments achieved while constructing these three vehicles signaled the beginning of a new technological era. In 1927 the UK Royal Navy presented the "Long-Range Gun With Lynx Engine" or "LARYNX", a monoplane capable of carrying a warhead of 114 kg, over a range of 480 km. Unlike the previous machines, it was fitted with radio control for the launch mode, after which the autopilot restrained it to fly on a preset course at a preset height to a preset range. The major achievement of this aircraft was to introduce a measure of radio control. Around the Second World War years, Great Britain decided to abandon the "cruise missile" concept and to concentrate on target aircraft with full-mission radio control. The British Royal Aircraft Establishment began test flight of target aircraft in 1921. Those studies gave birth to the "Fairey Queen" in 1933, and finally the "Queen Bee" target aircraft in 1934, see Fig. 1.2(a), which was the first non disposable target aircraft developed. It was designed for use during training missions, controlled by a human pilot via radio commands. This aircraft achieved flight autonomy of 480 km of distance, at over 160 km/h, flying as high as 5 km. At the same time, the Radioplane Company from the US developed the "RP4" an aircraft developed as gunnery practice target. Together the "Queen bee"
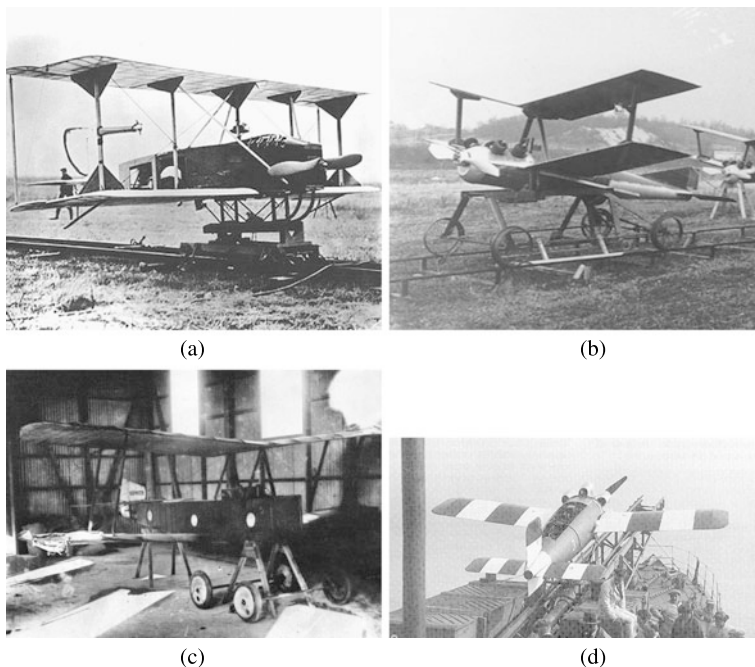
Fig. 1.1 Earliest UAVs systems: (**a**) "Aerial Torpedo"; (**b**) "Kettering Bug"; (**c**) "Aerial Target";
(**d**) "LARYNX"

and the "RP4" launched the development of radio control. Meanwhile, Nazi Germany launched the "Fieseler Fi 103", better known as the "V-1 Buzz Bomb", the first aircraft to use jet propulsion, see Fig. 1.2(b). Another major achievement during this period was the US Navy "Project Fox", an aircraft equipped with a RCA television camera on the nose. The vehicle was radio controlled from a TG-2 manned aircraft equipped with a television screen. This vehicle is considered an early experience of tele-operation control. An important development during the post war period was the use of UAVs as anti-radar systems. These aircraft, known as "Crossbows", were released from a parent aircraft to confuse the opponent radar system.

During the 1960s, the UAVs began to be used for reconnaissance purposes over enemy territory. The "Firebee", Fig. 1.3(a), carried a still camera, whose photographs were developed at base after the return of the UAV. The aircraft was recovered by a deployed parachute on returning to a suitable area for landing. This vehicle was harder to detect and harder to shoot down than manned reconnaissance aircraft, in addition, it eliminated diplomatic incidents upon the capture of a human pilot. The drone anti-submarine helicopter, known as "Gyrodyne DASH", see Fig. 1.3(b), was a specific and dedicated design conceived to carry torpedoes to attack enemy submarines. This system introduced for the first time the use of a rotor-craft UAV. UAVs specifically designed for reconnaissance and surveillance missions appeared during the 1970s. Also, the long-endurance characteristic became a pri-

Fig. 1.2 World War II UAVs: (**a**) "Queen Bee"; (**b**) "Fieseler Fi 103"

Fig. 1.3 1960s UAVs: (**a**) "Ryan Firebee"; (**b**) "Gyrodyne DASH"
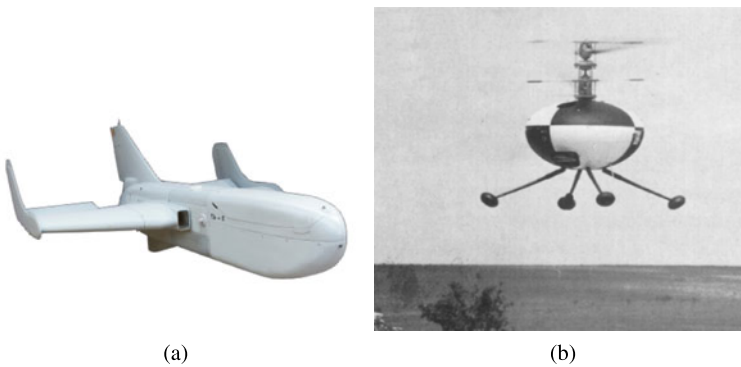
Fig. 1.4 1970s UAVs: (**a**) "MBLE Epervier"; (**b**) "Westland Wisp"

mary research. These developments were driven largely by the Cold War pressure. The "MBLE Epervier", Fig. 1.4(a), developed in Belgium, was powered by a small turbo jet engine, it was launched from a ramp mounted on a truck and the recovering was by parachute. Its flight profile was achieved through a pre-programmed autopilot. The payload of this vehicle consisted of a daylight camera or an infrared line-scan camera, whose data were processed once the UAV reached the ground sta-

(a)   (b)

(c)   (d)
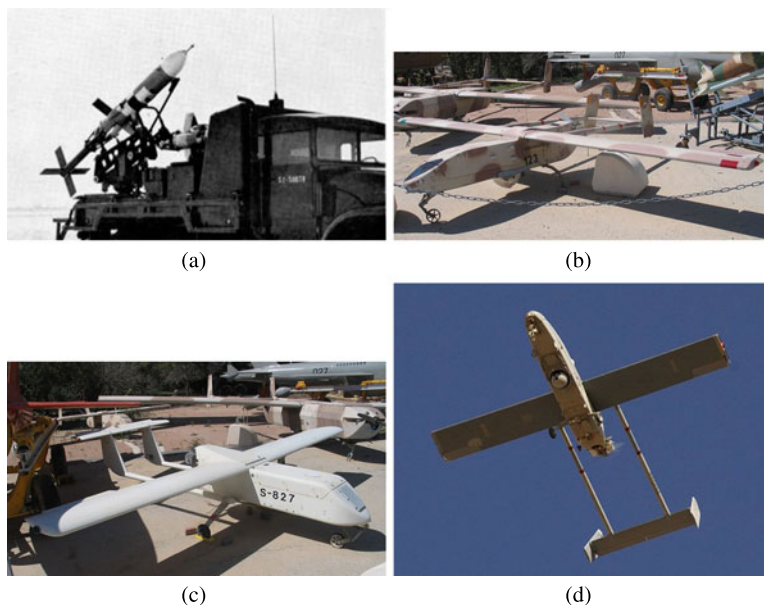
Fig. 1.5 1980s UAVs: (**a**) "Canadair CL-89"; (**b**) "IAI Scout"; (**c**) "Tadiran Mastiff"; (**d**)"Pioneer"

tion. Vertical take-off and landing (VTOL) vehicles appeared in shorter range operations. The "Westland Wisp" system, Fig. 1.4(b), a planar symmetric rotorcraft, carried a camera to send real-time video to the operator. This machines gave insight into the advantages of a hover capability during surveillance missions. During the 1980s, the UAVs surveillance and navigation systems became even more advanced The "Canadair CL-89", Fig. 1.5(a), provides non-real-time visual intelligence of enemy territory within an operating radius of 70 km. The guidance of this vehicle was achieved by a preset program controlling an autopilot supported by vertical and directional gyros and barometric sensing of speed and altitude. Navigation was by computer-based dead-reckoning. Latter versions of this aircraft achieved real-time video transmission and GPS aided navigation. Due to the increased accuracy and reliability of flight control systems, armies were by now looking to extend UAV operations to somewhat greater range (of order 100 km). The Israel Aircraft Industries built the "IAI Scout" UAV, see Fig. 1.5(b), a piston-engined aircraft with a 4 m wingspan made of fiberglass. Scout's fiberglass frame emitted an extremely low radar signature, which, coupled with the UAV small size, made it almost impossible to shoot down. The Scout UAV could transmit real-time, 360-degree surveillance via an omnidirectional camera in its central turret. This system, together with the similarly configured "Tadiran Mastiff", see Fig. 1.5(c), evolved to the "Pioneer" shown in Fig. 1.5(d), which remained in service with Israeli and US Forces until the mid 2000s. For guidance and control, the "Pioneer" employed an automatic flight control system and automatic tracking secure two-way data link. In the 1990s the increasing availability of the GPS freed the UAVs from their dependency on inaccurate

Fig. 1.6 1990s UAVs:
"Yamaha Rmax"

onboard navigation systems, based on computerized dead-reckoning. Another important achievement during this decade was the introduction of the first large-scale production rotary-wing UAV from Japan, the "Yamaha R50", and its enhanced version the "R.Max", see Fig. 1.6. This system, employed in agriculture, was successful in finding a niche market within which regulatory authorities allow it to operate. The 2000s decade saw a much increased use of UAVs in military roles. During these years, some systems, like the General Atomics "Predator B" shown in Fig. 1.7(a), began to amass operating hours measured in hundreds of thousands compared with just thousands in earlier decades. Another development in this decade was the empowerment of UAVs to carry armament while performing military reconnaissance missions. This was a modification of the "Predator B", presented under the name of "Reaper" UAV, see Fig. 1.7(b), which enabled an immediate attack response if needed. Although potentially more extensive than military, civilian operations have not come to fruition due to the perceived difficulty in ensuring separation between manned and unmanned systems. The beginning of the 2010s has seen a modest use of UAVs for civilian tasks, where the exceptional application of the "RMax" UAV is still considered as the most important one. This slow yet constant rate of development has been primarily due to airworthiness authorities, denying the use of suitable airspace to UAVs, until they are equipped with a reliable "sense-and-avoid" system. Nowadays, a considerable number of organizations are working on the investigation of bringing together the technologies to achieve the development of such system. In

(a)   (b)

Fig. 1.7 2000s UAVs: (**a**) "Predator B"; (**b**) "Reaper"

the USA, an encouraging development related to this subject was the landing and take-off of a "Predator" UAV at the Oshkosh General Aviation show in July 2009.

As happened in the last years, the continuous improvements and developments in electronic technology will increase the efficiency of many current systems. Surely, it will enable the development of UAVs with characteristics currently not even foreseen.

1.1.2 Applications

It is commonly believed that the use of UAVs is restricted to roles which are dull or dangerous for human pilots. Recent years and experiences have shown that covert, diplomatic, research and environmentally critical roles are also in the UAVs field of action. Among the civil roles of UAVs we can cite:

- *Images and video*: Aerial photography and video, news information, geological and archaeological survey, aerial photography for mapping.
- *Security*: pipeline security, powerline inspection, disaster control, search for missing persons, traffic and coastline monitoring, surveillance for illegal imports, incident control.
- *Environmental situations*: forestry fire detection, pollution monitoring and control, sampling and analysis of atmosphere for forecasting, fisheries protection.
- *Scientific research*: obtaining data from hazardous or remote locations.
- *Agriculture*: agriculture monitoring and spraying.

It is also well known that there exist several military applications where UAVs are being currently implemented.

1.1.3 UAVs Classification

UAVs are designed for different purposes, therefore, we can find in the literature many different ways to classify them. We present first a general classification of UAVs according to its range of action.

- **High-altitude long-endurance (HALE)**
  Systems capable of flying over 15000 m of altitude and more than 24 hr of endurance. Designed for performing long-range long-range reconnaissance and surveillance missions (Fig. 1.8).
- **Medium-altitude long-endurance (MALE)**
  Flying between 5000–15000 m of altitude, with a maximum of 24 hr endurance. Similar roles and characteristics to the HALE, but at shorter ranges (Fig. 1.9).
- **Medium-Range or Tactical UAV (TUAV)**
  For missions consisting of flights between 100 and 300 km. Smaller vehicles and operated within simpler systems than the HALE–MALE (Fig. 1.9).

Fig. 1.8 HALE UAV: Northrop Grumman's "Global Hawk"

- **Close-Range UAV**

  Operation range of 100 km. Probably the most prolific type of UAV, including roles as diverse as reconnaissance, target designation, airfield security, powerline inspection, crop-spraying and traffic monitoring, etc. (Fig. 1.10).

- **Mini UAV (MUAV)**

  UAV whose weight is in the order of 20 kg, operating at ranges of up to about 30 km.

- **Micro UAV (MAV)**

  UAVs having a wingspan no greater than 150 mm. Required for operations in urban environments, particularly within buildings. It is required to fly slowly, and preferably to hover. Research of this kind of UAV is being conducted into some

(a)   (b)
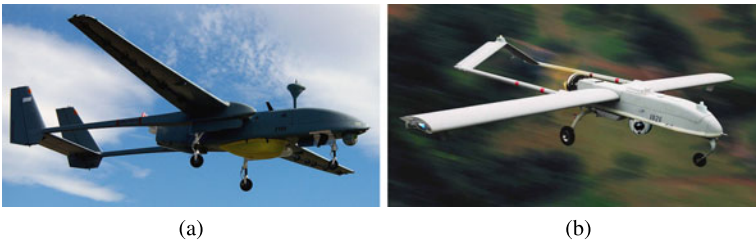
Fig. 1.9 MALE and TUAV UAVs: (**a**) IAI "Heron"; (**b**) AAI RQ-7 "Shadow"

(a)   (b)

Fig. 1.10 Close-Range and Mini UAVs: (**a**) Aeronautics Ltd "Aerolight"; (**b**) Insitu and Boeing's "ScanEagle"

(a)   (b)
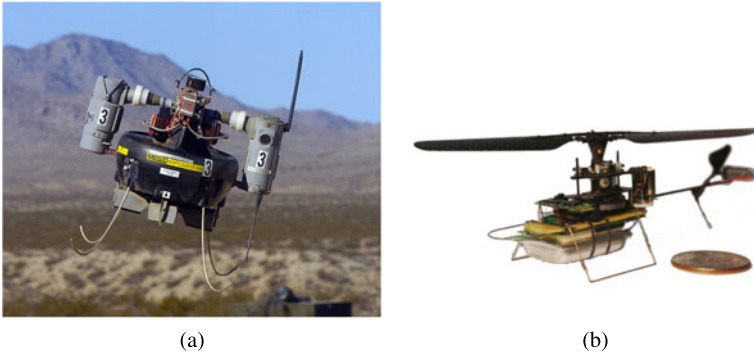
Fig. 1.11 Micro and Nano UAVs: (**a**) Honeywell's "RQ-16 T-Hawk"; (**b**) Prox Dynamics "Hornet 3"

less conventional configurations, such as flapping-wing aircraft. They are very vulnerable to atmospheric turbulence (Fig. 1.11).

- **Nano Air Vehicles (NAV)**
  Proposed to have a size of 10 mm, used in swarms for purposes such as radar confusion. They are also being proposed for ultra-short range surveillance, if camera, propulsion and control sub-systems can be made small enough to be implemented (Fig. 1.11).

- **Remotely piloted helicopter (RPH)**
  Aerial vehicle capable of performing vertical take-off and landing. They are normally used in missions that require hovering flight. Rotary wing aircraft are also less susceptible to air turbulence compared with fixed-wing aircraft of similar dimensions. Also known as *vertical take-off UAV* (VTUAV).

Moreover, in order to better describe their characteristics, UAVs can be classified as a function of their configuration. The four main categories are: (1) fixed-wing aircraft, with the advantage of high cruise speed and long endurance; (2) flapping-wing vehicles, flying like birds and insects; (3) blimps, such as long-endurance balloons; (4) rotary wings UAVs, called also Vertical Take-off and Landing (VTOL) rotorcraft, like conventional helicopters with the advantages of hovering capability and high maneuverability.

- **Fixed Wing**
  Fixed-wing UAVs in the civilian field are most used for long-distance, long-range and high-altitude missions. Commonly, they perform scientific applications such as meteorological reconnaissance and environmental monitoring. This kind of aircraft has benefited from having been developed for the military field, thus, it is equipped with similar systems in order to achieve its navigation tasks. Notable aircraft within this category are the "Aerosonde", developed by the Australian company Aerosonde Pty, Ltd., and the "Pathfinder", see Fig. 1.12, developed by the NASA.

Fig. 1.12 Fixed-wing
civilian UAV: NASA's
"Pathfinder"

- **Flapping wing**

  Flapping-wing UAVs try to reproduce the way birds or insects fly. Most of them
  are still under development. Belonging to the class of micro UAV, they have
  extremely low payload capability and low endurance. Among their interesting
  characteristics, flapping-wing UAVs have low power consumption and can per-
  form vertical take-off and landing. The "DelFly", see Fig. 1.13(a), is one of the
  flapping-wing UAVs of this kind. It has an embedded micro-camera, with the pur-
  pose of sending real-time video to the control station, this allows a remote pilot to
  control the UAV. The "DelFly" does not uses any sensors to ensure stabilization,
  and does not count with any guiding system.

  The FESTO's "SmartBird", see Fig. 1.13(b), is a powerful flight model with
  great aerodynamic qualities and extreme agility. The "SmartBird", which is in-
  spired by the herring gull, can start, fly, and land autonomously, with no additional
  drive mechanism. Its wings not only beat up and down, but also twist at specific
  angles. This is possible by an active articulated torsional drive unit, which in
  combination with a control system attains an unprecedented level of efficiency in
  flight operation. The "SmartBird" is considered as an energy-efficient technical
  adaptation of a model from nature.

  In spite of the evident advantages of this kind of UAV, the difficulties related
  to its construction and set-up have made that only a few these vehicles have been
  capable of flying.

- **Blimps**

  Blimps or "lighter-than-air" UAVs ensure lifting by means of their helium-filled
  ballonet, to enable long endurance. Since no energy is expended to lift the UAV,
  this saving can be used as power source for displacement actuators. An example
  of a blimp vehicle is the "C1000" from Skyships Ltd., a new generation of UAV
  aimed at low-level, local area missions. This airship has the capability to work
  safely at low levels, close to people and buildings. Equipped with limited navi-
  gation systems, most of the "C1000" vehicles flight manually, however, a few of
  them carry a GPS-based guiding system for navigation.

  Concerning lighter-than-air UAVs, FESTO is making the difference with the
  development of the "AirRay", the "AirJelly", and the "AirPenguin". "The Air-
  Ray" was developed in 2007, see Fig. 1.14(a). It is a remote-controlled hybrid
  construction, comprising a helium-filled ballonet and a flapping-wing driving
  mechanism. The AirRay's ballonet ensures hover flight, while the flapping-wing

(a)   (b)
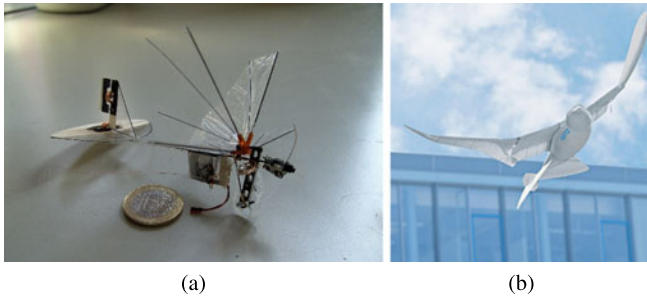
Fig. 1.13   Flapping-wing civilian UAV: (**a**) Technical University of Delft's "DelFly"; (**b**) FESTO's "SmartBird"

(a)   (b)

(c)
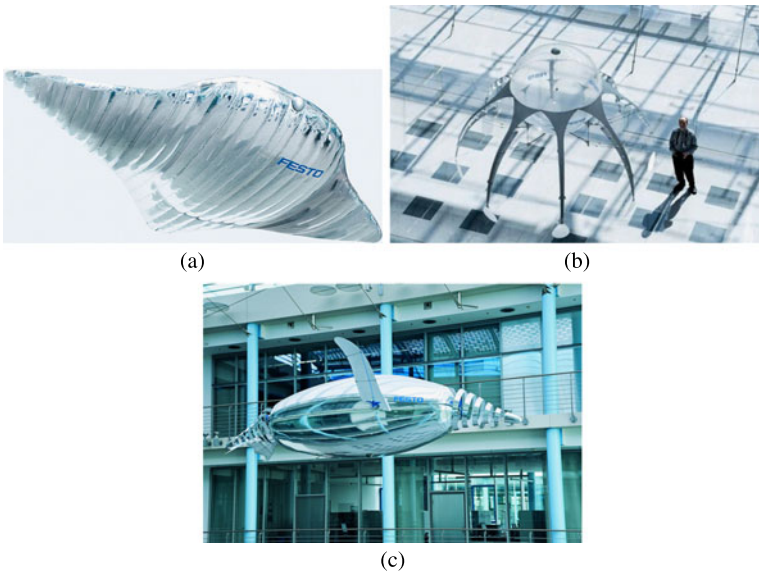
Fig. 1.14   Blimps UAVs: FESTO's (**a**) "AirRay"; (**b**) "AirJelly"; (**c**) "AirPenguin"

mechanism creates the force vectors needed for translational movements. Its autopilot system enables radio-manual control as well as autonomous hover. Another blimp UAV developed by FESTO in 2008 is the "AirJelly", see Fig. 1.14(b), also a radio-controlled vehicle. The "AirJelly" flies using an original propulsion system based on weight displacement. The system, comprised by eight tentacles, creates a force perpendicular to the ground plane. A pendulum that can be driven int the "x" and "y" directions is mounted inside the AirJelly's ballonet. Varying the pendulum position, the center of mass of the vehicle changes also, as a consequence, the vehicle moves in the corresponding direction. The "AirPenguin", shown in Fig. 1.14(c), is a prototype developed by FESTO in 2009. By using two

(a)   (b)
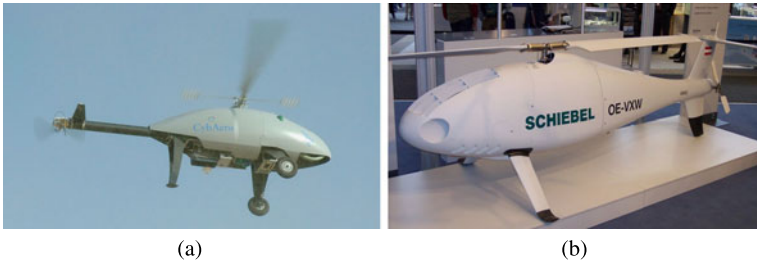
Fig. 1.15   Rotary wing UAVs: (**a**) CybAero's "Apid55"; (**b**) Schiebel Camcopter "S-100"

three-dimensional cones as the tail and the nose of the vehicle, the "AirPenguin" can be directed to any desired spatial orientation. The flapping wings are designed to bend in both directions allowing the forward and reverse motion. This vehicle flies smoothly, and is capable of avoiding obstacles and reaching any position by performing acrobatic maneuvers. In order to prove the versatility of this blimp vehicle, FESTO has tested a swarm-flight mode composed of three "AirPenguins", each one of them equipped with wireless modems and embedded autopilots. The AirPenguins recognize each other, estimating their distances to the transmitting stations. The controls implemented allow the AirPenguins to fly in a group without colliding. As an alternative, they can act synchronously as a group. This vehicle is considered as the most versatile and advanced blimp existent in the civil market.

- **Rotary wing**
  The common example of rotary wing UAVs are the helicopters: a two-rotor aircraft, with a main rotor giving the thrust and an anti-torque tail rotor. Nowadays, the Yamaha's RMAX helicopter is considered as the most advanced UAV commercially available, see Fig. 1.6. It is equipped with an attitude control system, characterized by its great capability to improve the tasks of hover flight, in addition, GPS-based autonomous navigation is possible. The RMAX was developed for the farming industry, specifically for crop dusting. However, its application range has grown widely, from environmental observation to construction site photography. In addition to the RMAX, a great number of autonomous helicopters have been developed.

  The CybAero's "Apid55" helicopter, Fig. 1.15(a), is a fully autonomous multipurpose VTOL UAV. It is equipped with several onboard systems (gyros, accelerometers, GPS, compass, infrared altimeter, barometric altitude) for achieving stabilization and navigation. The helicopter is designed to carry a wide range of sensors, such as stabilized cameras, laser scanners, infrared sensors and antennas. This payload, in combination with the wireless connection to a ground control station, enables the helicopter to perform tasks such as aerial photography, monitoring, search, and surveillance.

  Produced by the Austrian company Schiebel, the "Camcopter S-100," shown in Fig. 1.15(b), is controlled by a triple-redundant flight computer based on proprietary flight control methods and algorithms. Combining inertial navigation sys-

(a)   (b)

Fig. 1.16   Rotary wing UAVs: Swiss UAV and Saab (**a**) "NEO S-300"; (**b**) "KOAX X-240"

Fig. 1.17   Coaxial rotary
wing UAV: Skybotix's
"CoaX"

tems and a GPS modules, the autopilot ensures a precise navigation and stability
in all phases of flight. This UAV, capable of carrying a 34 kg payload for over
6 hours, is designed for missions such as monitoring of environmental and nat-
ural disasters, surveillance of large-scale commercial installations (oil pipelines,
offshore rigs, railways or powerlines), search and rescue, and aerial photogra-
phy. The "NEO S-300", see Fig. 1.16(a), developed by the company Swiss UAV
and Saab, features a traditional main and tail rotor assembly. It has an improved
system, focused on safety, by offering an integrated VTOL Parachute Rescue Sys-
tem (VPRS), and redundant data-links. The same companies have also developed
a coaxial helicopter: the "KOAX X-240", Fig. 1.16(b). The main advantage of
this configuration is the suppression of the tail rotor and the multiplication of the
total thrust force. The "KOAX X-240", belonging to the class of mini UAVs, is
well suited to perform missions in tight or confined environments. The vehicle's
payload is in the order of 8 kg and promises an autonomy of 90 minutes. A micro
coaxial helicopter called "CoaX", see Fig. 1.17, has been developed by Skybotix,
specifically designed for research and educational markets. It is equipped with
commercial sensors and processors. Two brushless motors conform the propul-
sion system, and two servo motors create the swash-plate tilting. The embedded
autopilot uses a home-made autopilot to stabilize the angular dynamics, and a
pressure sensor coupled with a down-looking sonar is used for altitude stabiliza-
tion. Three sonars and a color camera enable the guiding tasks, such as obstacle

(a)   (b)

Fig. 1.18 Convertibles VTOL UAVs: (**a**) "T-Wing"; (**b**) "Vertigo"

detection and avoiding, and relative navigation. Several companies and universities are developing rotary wing UAVs with unconventional designs. The most notable configurations are: convertibles VTOL, quad-rotors, six-rotors and eight-rotors. Contained in the category of micro UAVs, they are small enough to fly indoors and in places where other autonomous helicopters or aircraft cannot reach. This characteristic gives them an important place in the civil field. Generally, these kinds of vehicle are electric powered, which makes them safer to operate in high populated environments. With the exception of convertibles VTOL, their design searches to simplify the mechanics necessary to the generation of the control forces and torques.

Convertibles VTOL aircraft, such as the "T-Wing" by the University of Sydney, Australia, and the "Vertigo" by the Institut Supérieur de l'Aéronautique et de l'Espace (SUPAERO), France, were specially developed to achieve a transition from vertical flight to horizontal flight. The T-Wing shown in Fig. 1.18(a) is a VTOL UAV capable of both wing-born horizontal flight and propeller-born vertical mode flight, including hover and descent. The "Vertigo", see Fig. 1.18(b), is a small aircraft that uses coaxial propellers as its propulsion system, and possesses control surfaces in order to control the roll, pitch and yaw angles, can be used for missions like observation of urban environment. The designs using multiple rotors allow the simplification of forces generation and torque creation, as mentioned previously. The quad-rotor design allows the elimination of the swashplate, a very complex mechanical structure, by taking advantage of its four motors and the implementation of differential control. Furthermore, it automatically eliminates the gyroscopic torques created by the spinning motors. This interesting configuration allows each individual rotor to have a smaller diameter than the equivalent ordinary helicopter rotor, for a similar vehicle size, allowing them to store less kinetic energy during flight. For small scale UAVs, this makes the vehicles safer to interact within close proximity. By enclosing the rotors within a frame, the rotors can be protected during collisions, permitting flights indoors and in obstacle-dense environments, with low risk of damaging the vehicle, its operators, or its surroundings.

Concerning the most notable quad-rotor systems, the German company Ascending Technologies (AscTec) has developed a quad-rotor UAV called "Peli-

Fig. 1.19   Quad-Rotor UAVs: (**a**) AscTec "Pelican"; (**b**) Aeryon Labs "Scout"

Fig. 1.20   Multi-Rotor UAVs: (**a**) Parrot's "AR.Drone"; (**b**) DraganFly "Draganflyer X6"

can", see Fig. 1.19(a). This vehicle is optimized to carry up to 500 grams of payload and has demonstrated extraordinary capacities for performing autonomous navigation tasks using inertial sensors, laser range finder and video cameras. The "Aeryon Scout", see Fig. 1.19(b), is a quad-rotor UAV designed by the Canadian company Aeryon Labs. This vehicle can operate up to 3 km from the user, at flying speeds of up to 50 km/h. Unlike many other UAVs in this class, the Scout can tolerate winds of up to 80 km/h. Its payload consists of high resolution, day–night cameras. An interesting characteristic of this UAV is that all communications between the vehicle and the base station are digital and encrypted, which prevents hijacking and video interception. Finally, another interesting quad-rotor is the "AR.Drone" rotorcraft, see Fig. 1.20(a), designed specifically for the gaming market by the French company Parrot. The main breakthrough of this vehicle is the incredibly cost reduction of each system. This rotorcraft presents high performances since it uses an optimized embedded system that deals with the vehicle stabilization and image processing. An example of an innovative six-rotor design is the "Draganflyer X6", see Fig. 1.20(b), from Draganfly Innovations. The vehicle's six rotors are arranged as three counter-rotating offset pairs mounted at the ends of the three arms, with matched sets of counter-rotating rotor blades. The control of the rotorcraft is obtained by differential thrust from the three pairs of motors. By using two motors in each arm, the vehicle obtains almost the double of total thrust, without increasing the size of the aircraft. Since each arm has two

                    (a)   (b)

Fig. 1.21   Eight-Rotor UAVs: (**a**) DraganFly "Draganflyer X8"; (**b**) AscTec "Falcon 8"

rotor blades turning in opposite direction, the gyroscopic effect is also automatically compensated. Moreover, this configuration increases the flight efficiency, and the vehicle response against perturbations.

UAVs equipped with eight rotors exist in different types, where the motors arrangement is decided by the vehicle's manufacturer according to their specific goals. The "Draganflyer X8" shown in Fig. 1.21(a), from Draganfly Innovations, uses an innovative tilted eight-rotor design. The eight rotors are arranged as four counter-rotating offset pairs, mounted at the ends of four carbon fiber arms. The four sets of matched counter-rotating rotor blades provide differential thrust from four equally spaced points, which allows the multi-rotor UAV to maneuver with high precision. In this configuration, the rotor blades of the counter-rotating pair eliminate perfectly the momentum created by each other. Finally the eight electric motors allow the "Draganflyer X8" to carry payloads of 1 kg approximatively. Using a GPS-INS based autopilot, the vehicle navigates completely autonomous.

The "Falcon 8", see Fig. 1.21(b), developed by AscTec is equipped with eight rotors, arranged to the form a "delta". This aircraft has high levels of flight stability, dealing with winds up to 10 m/s. One of the most interesting characteristics of the Falcon 8 consists of the fact that if one of the rotor/motor combinations fail during flight, the aircraft is still able to continue flying with normal functionality, this is obtained since each rotor/motor is redundant in the system. The "Falcon 8" is able to hold its position using GPS information, in addition, autonomous navigation is available via way-point planning.

1.2 State of the Art

With the objective of developing a UAV capable of performing tasks in an autonomous manner, a great effort has been made for designing an effective sensor suit for estimating the states of the vehicle in a fast and accurate way, as well as for sensing its surrounding environment. Once the variables representing the behavior of the vehicle and its position in the world are known, it is possible to develop

control strategies for stabilizing the vehicle during flight, so as to enable it to perform tasks autonomously. Initially, for estimating the vehicle's states a lot of work focused around the use of inertial systems and unreliable methods such as "dead-reckoning", obtaining in general poor results. Also, a combination of inertial and GPS systems has been studied. Unfortunately, this sensing method is restricted to places where the positioning system signals are available.

Recently, research groups have been attracted by the implementation of imaging sensors for completing the sensor suit of an aerial vehicles. Due to the optical sensor capacity for sensing its surroundings, new techniques can be developed, allowing the vehicle to perform the tasks of relative positioning and autonomous navigation. According to the different kinds of mission and the environments where the UAV must interact, solutions based on monocular vision, stereo vision and even multiple views are being proposed.

A vision-based approach for performing an autonomous landing of a helicopter was presented in [72]. The vision system was used for target detection, while a combination of vision and GPS measurements allowed autonomous navigation. The helicopter updates its landing target parameters based on vision and uses an on-board behavior-based controller to follow a path to the landing site. Once a ground target has been identified on the image, the position of the vehicle that causes the respective image formation can be deduced. The inverse process is also valid. Given a desired image and the current image of the ground target, a precise control can be constructed, in order to displace the vehicle to the position that allows the formation of the desired image. This vision-based position control is called *Visual Servoing*. First, a relationship between the image-based task and the actuators is modeled. Then, the imaging measurements are used to control the robot directly.

Another visual servoing technique is presented in [39], where the authors used a vision-based control technique for stabilizing the position of a quad-rotor over an on-ground target formed by four black circles. In this approach, the desired position of the quad-rotor is deduced using a specific position configuration of the four circles in the image plane. Following a similar approach, a visual servo control based on the color detection of four points was presented in [66]. Real-time experiments have shown the position stabilization of the vehicle, with respect to the desired configuration of the points in the image plane.

A vision-based strategy permitting a helicopter to detect and track powerlines for inspection was presented in [25]. The goal of this study was to achieve steady flight near the powerlines, as well as performing missions consisting of powerlines following. By taking advantage of a previously known model of a powerline cable, the authors developed a vision algorithm allowing the helicopter to estimate its position with respect to the powerlines.

With the purpose of performing autonomous navigation and tracking, a system that combines information from an imaging sensor with GPS measurements was presented in [73]. The objective of such study is to locate features that correspond to a building's window, once detected, those windows are tracked over time. The vision algorithm estimates high-level velocity commands and sends them to the helicopter main controller, which is then able to command the helicopter to track them.

They used a gas-powered autonomous helicopter with an embedded computational payload formed by a PC-104 computer, a group of inertial sensors and a Novatel RT-2 DGPS. This work benefited from artificial or human built structures, using them as features of interest. However, natural features are also suitable for performing vision-based state estimation and control.

In [65] the authors developed a vision-based navigation control system for a glider, using a monocular imaging sensor. The main contribution of this work is that the vehicle does not depend on inertial sensors for states estimation. Overall control and guidance is achieved by using only the information recovered by the camera. A fast pattern matching is applied to detect an artificial target in the image plane, then an extended Kalman filter computes the estimates of the required states for controlling the vehicle. The implementation of this kind of strategy was possible since they used a fixed-wing aircraft, which has better natural stability than a rotary wing aircraft. Clearly, the probability of obtaining a similar performance over a quad-rotor rotorcraft is almost zero.

Natural features were used in [50] to design a vision-based landing system, with the purpose of enabling the aerial robot for identifying and avoiding hazardous entities, such as steep slopes or large rocks. The authors used images from a single camera in a structured-from-motion algorithm to generate a dense elevation map of the area below the helicopter, making possible the detection of a safe landing site. Autonomous flight was achieved using a hierarchical control architecture, which can be also described as a combination of high-level and low-level controllers. Using this approach, the helicopter was able to land autonomously in an unknown terrain.

A vision system for the Yamaha Rmax helicopter, allowing autonomous navigation, automatic searching of on-ground objects and object tracking was presented in [59]. The presented algorithm is capable of detecting previously known targets in order to reduce the processing time requirements. An extended Kalman filter is implemented to perform a data fusion between inertial and visual measurements, allowing a full estimation of the vehicle's states. The authors' strategy was tested in real-time experiments. For performing the image processing, a Pentium III PC was installed onboard the aerial vehicle.

Object detection and visual servoing approaches are suitable for specific tasks where the main goal is performing a hover flight or landing. However, for navigation purposes, more elaborate solutions must be developed. Vision-based odometry and Simultaneous Localization and Mapping (SLAM) are being used to fill the gap left by other vision-based approaches, in terms of relative positioning and mapping. Since single-camera systems offer a good solution concerning weight and processing time, monocular odometry and SLAM are the most common approaches in vision-aided navigation systems for UAVs.

A vision-based strategy for localizing a UAV by means of visual odometry (VO) and SLAM was presented in [23]. Its method relies on the detection of natural landmarks, eliminating the need for artificial visual cues with previously known positions. The monocular VO is used as a backup system when the accuracy of GPS is reduced to critical levels. In addition, an homography-based technique is used

to compute the UAV relative translation and rotation by means of the images gathered by the onboard camera. In order to reduce the impact of cumulative errors in odometry-based position estimation approaches, the VO is integrated into a SLAM scheme.

In [13] the authors implemented a SLAM algorithm for the localization of a quad-rotor UAV, using an onboard monocular camera pointing downwards. A state-of-the-art visual SLAM algorithm tracks the pose of the camera, while, simultaneously, building an incremental map of the surrounding region. Using the pose measurements, a controller stabilizes the vehicle at a desired setpoint, making possible the maneuvers of take-off, hovering, setpoint following or landing.

A visual and inertial system for a quad-rotor helicopter is proposed in [4], with the purpose of enabling navigation in unknown, indoor, GPS-denied environment. This approach allows fast environment mapping and obstacle avoidance. The ego-motion of the rotorcraft is estimated from a group of tracked features. An extended Kalman filter integrates inertial and ego-motion measurements.

In [26] the authors present a general discussion concerning the different techniques available for the design of vision-aided navigation strategies using monocular systems. Aspects concerning algorithms for detection and tracking, visual servoing and control, and the integration on a global visual SLAM approach are treated on detail. The addressed techniques are evaluated in real-time experiments, and their performance is compared with the performance of a stereo vision system.

Without simplifying assumptions or the use of previously known landmarks, a single-camera system is capable of motion estimation up to a scale factor. In addition, estimations concerning displacements over the optical axis requires heavy computational process. A stereo camera system, on the other hand, allows computing the depth of a group of tracked landmarks directly from the previously known stereo-camera's geometry. This allows a direct estimation of the three-dimensional motion of the UAV without the need of assumptions for simplifying the problem.

A stereo vision system with a baseline of 12 cm is presented in [82], with the purpose of performing an autonomous landing of a small helicopter. Since the cameras are looking downward, the disparity chart between the right and the left image permits to estimate the height of the helicopter during flights. The z-axis accelerometer is fused with the altitude measurement in order to estimate the vertical velocity. Once, position and velocity are computed, a two-stage control strategy is performed to enable the landing task.

In [3], the authors propose a three-dimensional VO system for a quad-rotor UAV based on stereo vision. The pose of the stereo-rig is estimated at each time-step, and the integration of two successive poses allows computing the relative displacement of the vehicle. A dense-stereo procedure is applied to ensure a good pose estimation, then, the resulting measurement serves as an input for a robust estimation based on a RanSaC algorithm. Finally, position drift is compensated via a SLAM approach.

Stereo vision systems imitate the three-dimensional reconstruction process carried out by animals, birds and humans. However, it is possible to use two or more views, in different configurations, in order to improve the position estimation task.

A novel two-camera method for estimating the full six-degrees-of-freedom pose of a quad-rotor helicopter is presented in [6]. One of these cameras is located on-board the helicopter, and the other camera is located on the ground, thus, both cameras can see each other. Real-time experiments, consisting of vision-based stabilization, have shown the effectiveness of this approach.

In [34] the authors presented a mixed stereoscopic vision system; consisting of a fish-eye camera and a perspective camera, with the purpose of performing altitude estimation. Since there exists a homography between the two views, and the stereoscopic sensor had previously been calibrated, the proposed algorithm searches the altitude which verifies this homography.

Vision systems consisting of omnidirectional cameras have been studied also. A system for a helicopter, consisting of an omnidirectional camera, was presented in presented in [47]. The experimental results shown that the measurements obtained from such system are suited to perform the tasks of navigation and obstacle avoidance. A similar approach is presented in [32], where the authors discuss the characteristics and advantages of a catadioptric vision sensor. They propose a method consisting of separating the sky from the earth, in order to detect the horizon, which allows the computation of roll and pitch angles. The algorithm performance was tested using data recorded from a fixed-wing aircraft during flight.

Optical flow techniques are commonly implemented in the development of autonomous navigation systems for UAVs. In [46] the authors address the problem of navigating a UAV safely through urban canyons, by using two obstacle-avoidance techniques based on stereo vision and optic flow. Optic flow is evaluated for its ability to produce a centering response between obstacles, and stereo vision is evaluated for detecting obstacles to the front. The combination of these two techniques was evaluated in real-time experiments. The proposed methodology allowed a UAV to detect an obstacle to the front, and optic flow allowed it to turn away from obstacles to the side.

1.3 Problem Statement

If significant advances in terms of UAVs stabilization have been made in recent years, it is also true that there is a lot of work to do in order to provide an effective autonomy to these vehicles. A basic requirement for UAVs consists of robust and autonomous hover flight and navigation. Previous tasks strongly depend on the UAV's capacity for estimating its own states and sensing its surrounding environment.

The common choice for estimating the position and velocity of a UAV is GPS, which is available in most of outdoor environments. Unfortunately, when UAVs are required to accomplish tasks through indoors or urban environments, GPS signals may be noisy or even unavailable, which leads to a poor or undesirable performance of the vehicle. Furthermore, achieving autonomous tasks becomes even more complicated if the vehicle evolves in GPS-denied and unstructured environments.

In order to achieve a fully functional UAV, the combination of different research areas must be considered, such as control, mechanics, computer vision and embedded electronic systems. In the present research work, imaging, inertial and altitude sensing systems are being implemented, enabling a quad-rotor UAV for performing the tasks of relative positioning and navigation.

1.4 Contributions

The main contributions of this book are detailed next.

The first contribution consists of the development of the quad-rotor platform itself. An experimental system consisting of the aerial vehicle and a supervisory ground station was designed, developed and improved, making real-time experiments to verify its performance. The efforts in this subject gave as result a quad-rotor robotic helicopter well suited for performing hovering flight and navigation tasks. An embedded sensor suit was specifically developed for estimating the vehicle's angular dynamics and attitude. It consists of an inertial measurement unit that provides the vehicle's Euler angles, in addition to three analog rate gyros for measuring the platform angular rate. An ultrasonic sensor and an atmospheric pressure sensor are also installed, in order to estimate helicopter altitude at low and high flights. It has been proved that the vehicle is easy to control, robust with respect to external perturbations and with an adequate time of autonomy. In addition, it has also the important characteristic of being able to lift a considerably amount of payload, which enables the vehicle to carry a considerable group of embedded electronics and sensors. The supervisory ground station proved to be effective for flight supervision, performing manually guided flights and vision-based control.

An important subject is the development of an experimental platform as stable as possible. The second contribution concerns the development and evaluation of control strategies for the quad-rotor UAV. An embedded control system for improving attitude stabilization of a quad-rotor mini UAV was proposed an tested in real-time applications. This control strategy uses low cost components and includes an extra control loop based on motor armature current feedback. The additional control loop significantly improves the performance of the quad-rotor attitude stability, eliminating the pitch and roll errors with respect to a horizontal plane. The technique proposed results in a controller that is robust with respect to external disturbances as has been observed experimentally. In addition, a comparison of three well known control techniques was conducted in a real-time application, with the main objective of obtaining the best control strategy for stabilizing the quad-rotor.

The third contribution concerns the development of two vision-based strategies for stabilizing the quad-rotor during flight. The system developed for this application consists of a calibrated camera onboard the UAV, an artificial marker placed on ground, an imaging processing and a control algorithms running on the supervisory ground station PC, and a wireless link between the helicopter and the supervisory ground station. A pair of imaging algorithms for detecting the three-dimensional position and translational velocity of the quad-rotor with respect to

an artificial landmark placed on ground were introduced. The visual information extracted from those methods allows the development of control strategies for performing autonomous flights consisting of position hold and forward flight at constant speed.

The fourth contribution concerns the development of an imaging, inertial and altitude sensing system, which enables the helicopter to fully estimate its states without using GPS or artificial visual landmarks. A sensor fusion strategy was implemented to combine visual odometry (VO) measurements, IMU-based acceleration data, and altitude sensor signals, with the purpose of providing accurate estimations of the states describing the translational behavior of the platform. The sensing system was tested in real-time flight experiments, where the three-dimensional trajectory described by the UAV was reconstructed satisfactorily. In addition, the search of an efficient state observer for obtaining the quad-rotor translational dynamics in an accurate way was also addressed. A Luenberger observer, a Kalman filter and a complementary filter were implemented and compared in real-time experiments consisting of an autonomous position hold.

1.5 Book Outline

The outline of this book is as follows. In Chap. 2, the modeling of a quad-rotor rotorcraft is presented. Chapter 3 presents the development of a supervisory ground station and three quad-rotor experimental platforms. A hierarchical control strategy is also introduced and some autonomous hover flight experiments are shown. The design of an embedded control strategy for improving the attitude stabilization of a quad-rotor UAV is presented in Chap. 4. This technique includes an extra control loop and is based on motor armature current feedback. Chapter 5 deals with the required theoretical background for the implementation of imaging system for estimating the sates of a UAV. Two different vision-based strategies for stabilizing a quad-rotor during flight are presented in Chap. 6. Also, a comparison of three control strategies is addressed, with the purpose of validating the most effective approach for stabilizing the vehicle when using visual feedback. Theoretical and practical aspects are detailed in this chapter, as well as the real-time experiments performed. With the purpose of enabling a UAV to autonomously perform take-off, relative positioning, navigation and landing when evolving in unstructured, indoors, and GPS-denied environments, Chap. 7 presents a stereo imaging, inertial and altitude sensing system for the quad-rotor. A comparison study between a Luenberger observer, a Kalman filter and a complementary filter is also addressed, with the purpose of validating the most effective approach for combining the different sensing technologies. Finally, conclusions and future work concerning the previously presented topics are given in Chap. 8.

Chapter 2
Modeling the Quad-Rotor Mini-Rotorcraft

The complete dynamics of an aircraft, taking into account aero-elastic effects, flexibility of the wings, internal dynamics of the engine and the whole set of changing variables are quite complex and somewhat unmanageable for the purposes of control. Therefore, it is interesting to consider a simplified model of an aircraft formed by a minimum number of states and inputs, but retaining the main features that must be considered when designing control laws for a real aircraft.

This chapter deals with the modeling of a quad-rotor rotorcraft, and is organized as follows. Section 2.1 gives a general overview of the quad-rotor aerial vehicle and its operation principle. Next, Sect. 2.2 deals with the quad-rotor modeling, presenting two different approaches: Euler–Lagrange in Sect. 2.2.1 and Newton–Euler in Sect. 2.2.2. Subsequently, it is shown in Sect. 2.2.3 how to derive Lagrange's equations from Newton's equations. Section 2.2.4 presents a Newton–Euler modeling for an "X-Flyer" quad-rotor configuration. Finally, some concluding remarks are presented in Sect. 2.3.

2.1 The Quad-Rotor Mini-Rotorcraft

The quad-rotor mini-rotorcraft is controlled by the angular speeds of four electric motors as shown in Fig. 2.1. Each motor produces a thrust and a torque, whose combination generates the main thrust, the yaw torque, the pitch torque, and the roll torque acting on the quad-rotor. Conventional helicopters modify the lift force by varying the collective pitch. Such aerial vehicles use a mechanical device known as swashplate. This system interconnects servomechanisms and blade pitch links in order to change the rotor blades pitch angle in a cyclic manner, so as to obtain the pitch and roll control torques of the vehicle. In contrast, the quad-rotor does not have a swashplate and has constant pitch blades. Therefore, in a quad-rotor we can only vary the angular speed of each one of the four rotors to obtain the pitch and roll control torques.

From Fig. 2.1 it can be observed that the motor M_i (for $i = 1, \ldots, 4$) produces the force f_i, which is proportional to the square of the angular speed, that is, $f_i = k w_i^2$.

Fig. 2.1 The quad-rotor
control input
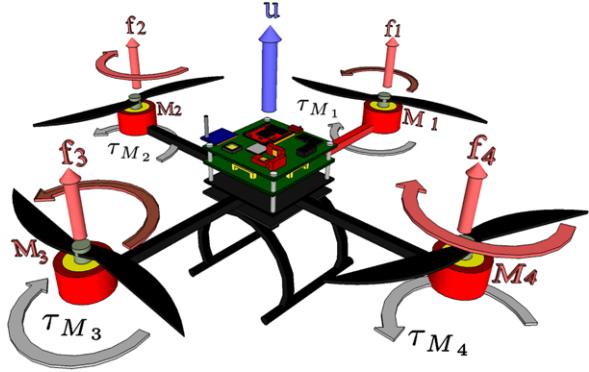
Given that the quad-rotor's motors can only turn in a fixed direction, the produced force f_i is always positive. The front (M_1) and the rear (M_3) motors rotate counter-clockwise, while the left (M_2) and right (M_4) motors rotate clockwise. With this arrangement, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight. The main thrust u is the sum of individual thrusts of each motor. The pitch torque is a function of the difference $f_1 - f_3$, the roll torque is a function of $f_2 - f_4$, and the yaw torque is the sum $\tau_{M_1} + \tau_{M_2} + \tau_{M_3} + \tau_{M_4}$, where τ_{M_i} is the reaction torque of motor i due to shaft acceleration and blades drag. The motor torque is opposed by an aerodynamic drag τ_{drag}, such that

$$I_{\mathrm{rot}}\dot{\omega} = \tau_{M_i} - \tau_{\mathrm{drag}} \qquad (2.1)$$

where I_{rot} is the moment of inertia of a rotor around its axis. The aerodynamic drag is defined as

$$\tau_{\mathrm{drag}} = \frac{1}{2}\rho A v^2 \qquad (2.2)$$

where ρ is the air density, the frontal area of the moving shape is defined by A, and v is its velocity relative to the air. In magnitude, the angular velocity ω is equal to the linear velocity v divided by the radius of rotation r

$$\omega = \frac{v}{r} \qquad (2.3)$$

The aerodynamic drag can be rewritten as

$$\tau_{\mathrm{drag}} = k_{\mathrm{drag}}\omega^2 \qquad (2.4)$$

where $k_{\mathrm{drag}} > 0$ is a constant depending on the air density, the radius, the shape of the blade and other factors. For quasi-stationary maneuvers, ω is constant, then

$$\tau_{M_i} = \tau_{\mathrm{drag}} \qquad (2.5)$$

Forward pitch motion is obtained by increasing the speed of the rear motor M_3 while reducing the speed of the front motor M_1. Similarly, roll motion is obtained using the left and right motors. Yaw motion is obtained by increasing the torque of the

Fig. 2.2 Pitch, roll and yaw torques of the quad-rotor
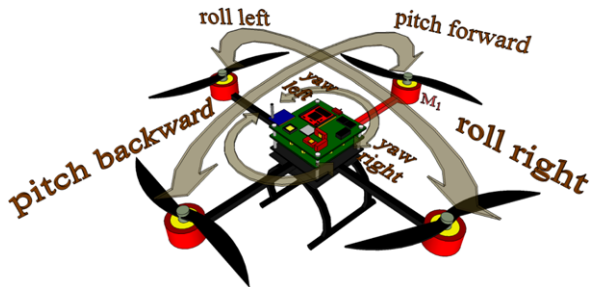
front and rear motors (τ_{M1} and τ_{M3}, respectively) while decreasing the torque of the lateral motors (τ_{M2} and τ_{M4}, respectively). Such motions can be accomplished while maintaining the total thrust constant, see Fig. 2.2.

2.2 Quad-Rotor Dynamical Model

The quad-rotor model is obtained by representing the aircraft as a solid body evolving in a three dimensional space and subject to the main thrust and three torques: pitch, roll and yaw.

2.2.1 Euler–Lagrange Approach

Let the generalized coordinates of the rotorcraft be expressed by

$$\mathbf{q} = (x, y, z, \psi, \theta, \phi) \in \mathbb{R}^6 \tag{2.6}$$

where $\boldsymbol{\xi} = (x, y, z) \in \mathbb{R}^3$ denotes the position vector of the center of mass of the quad-rotor relative to a fixed inertial frame $\mathscr{I}$. The rotorcraft's Euler angles (the orientation of the rotorcraft) are expressed by $\boldsymbol{\eta} = (\psi, \theta, \phi) \in \mathbb{R}^3$, ψ is the yaw angle around the z-axis, θ is the pitch angle around the y-axis and ϕ is the roll angle around the x-axis (see [33] and [5]). An illustration of the generalized coordinates of the rotorcraft is shown in Fig. 2.3. Define the Lagrangian

$$L(\mathbf{q}, \dot{\mathbf{q}}) = T_{\text{trans}} + T_{\text{rot}} - U \tag{2.7}$$

where $T_{\text{trans}} = \frac{m}{2} \dot{\boldsymbol{\xi}}^{\mathrm{T}} \dot{\boldsymbol{\xi}}$ is the translational kinetic energy, $T_{\text{rot}} = \frac{1}{2} \boldsymbol{\Omega}^{\mathrm{T}} I \boldsymbol{\Omega}$ is the rotational kinetic energy, $U = mgz$ is the potential energy of the rotorcraft, z is the rotorcraft altitude, m denotes the mass of the quad-rotor, $\boldsymbol{\Omega}$ is the vector of the angular velocity, I is the inertia matrix and g is the acceleration due to gravity. The angular velocity vector $\boldsymbol{\omega}$ resolved in the body-fixed frame is related to the generalized velocities $\dot{\boldsymbol{\eta}}$ (in the region where the Euler angles are valid) by means of the standard kinematic relationship [38]

Fig. 2.3 The quad-rotor in an inertial frame. f_1, f_2, f_3, f_4 represent the thrust of each motor, ψ, θ and ϕ represent the Euler angles, and u is the main thrust
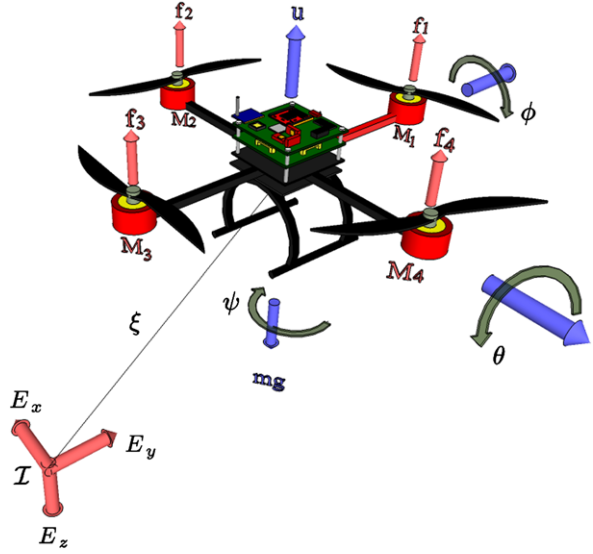
$$\boldsymbol{\Omega} = W_\eta \dot{\boldsymbol{\eta}} \tag{2.8}$$

where

$$W_\eta = \begin{bmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\phi & \cos\phi & 0 \\ \cos\theta\cos\phi & -\sin\phi & 0 \end{bmatrix} \tag{2.9}$$

then

$$\boldsymbol{\Omega} = \begin{bmatrix} \dot{\phi} - \dot{\psi}\sin\theta \\ \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \\ \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \end{bmatrix} \tag{2.10}$$

Define

$$\mathbb{J} = \mathbb{J}(\boldsymbol{\eta}) = W_\eta^\mathrm{T} I W_\eta \tag{2.11}$$

where

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{2.12}$$

so that

$$T_{\mathrm{rot}} = \frac{1}{2}\dot{\boldsymbol{\eta}}^\mathrm{T}\mathbb{J}\dot{\boldsymbol{\eta}} \tag{2.13}$$

Thus, the matrix $\mathbb{J} = \mathbb{J}(\boldsymbol{\eta})$ acts as the inertia matrix for the full rotational kinetic energy of the quad-rotor, expressed directly in terms of the generalized coordinates $\boldsymbol{\eta}$.

The model of the full rotorcraft dynamics is obtained from Euler–Lagrange equations with external generalized forces

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathscr{L}}{\partial \mathbf{q}} = \begin{bmatrix} \mathbf{F}_\xi \\ \tau \end{bmatrix} \tag{2.14}$$

where $\mathbf{F}_\xi = R\hat{\mathbf{F}} \in \mathbb{R}^3$ is the translational force applied to the rotorcraft due to main thrust, $\tau \in \mathbb{R}^3$ represents the yaw, pitch and roll moments and R denotes the rotational matrix. $R(\psi, \theta, \phi) \in SO(3)$ represents the orientation of the aircraft relative to a fixed inertial frame:

$$R = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\theta s_\phi s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{2.15}$$

where c_θ stands for $\cos\theta$ and s_θ for $\sin\theta$. From Fig. 2.1, it follows that

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \tag{2.16}$$

where u is the main thrust directed out of the bottom of the aircraft and expressed as

$$u = \sum_{i=1}^{4} f_i \tag{2.17}$$

and, for $i = 1, \ldots, 4$, f_i is the force produced by motor M_i, as shown in Fig. 2.1. Typically $f_i = k\omega_i^2$, where k_i is a constant and ω_i is the angular speed of the ith motor. The generalized torques are thus

$$\tau = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \triangleq \begin{bmatrix} \sum_{i=1}^{4} \tau_{M_i} \\ (f_2 - f_4)\ell \\ (f_3 - f_1)\ell \end{bmatrix} \tag{2.18}$$

where ℓ is the distance between the motors and the center of gravity, and τ_{M_i} is the moment produced by motor M_i, for $i = 1, \ldots, 4$, around the center of gravity of the aircraft.

Since the Lagrangian contains no cross terms in the kinematic energy combining $\dot{\xi}$ with $\dot{\eta}$, the Euler–Lagrange equation can be partitioned into dynamics for ξ coordinates and η coordinates. The Euler–Lagrange equation for the translational motion is

$$\frac{d}{dt}\left[\frac{\partial L_{\mathrm{trans}}}{\partial \dot{\xi}}\right] - \frac{\partial L_{\mathrm{trans}}}{\partial \xi} = \mathbf{F}_\xi \tag{2.19}$$

then

$$m\ddot{\xi} + mg\mathbf{E}_z = \mathbf{F}_\xi \tag{2.20}$$

As for the η coordinates, it can be written

$$\frac{d}{dt}\left[\frac{\partial L_{\mathrm{rot}}}{\partial \dot{\eta}}\right] - \frac{\partial L_{\mathrm{rot}}}{\partial \eta} = \tau \tag{2.21}$$

or

$$\frac{d}{dt}\left[\dot{\eta}^{\mathrm{T}}\mathbb{J}\frac{\partial\dot{\eta}}{\partial\eta}\right] - \frac{1}{2}\frac{\partial}{\partial\eta}\left(\dot{\eta}^{\mathrm{T}}\mathbb{J}\dot{\eta}\right) = \tau \qquad (2.22)$$

Thus one obtains

$$\mathbb{J}\ddot{\eta} + \dot{\mathbb{J}}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}\left(\dot{\eta}^{\mathrm{T}}\mathbb{J}\dot{\eta}\right) \qquad (2.23)$$

Defining the Coriolis-centripetal vector

$$\bar{V}(\eta, \dot{\eta}) = \dot{\mathbb{J}}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}\left(\dot{\eta}^{\mathrm{T}}\mathbb{J}\dot{\eta}\right) \qquad (2.24)$$

one writes

$$\mathbb{J}\ddot{\eta} + \bar{V}(\eta, \dot{\eta}) = \tau \qquad (2.25)$$

but $\bar{V}(\eta, \dot{\eta})$ can be expressed as

$$\bar{V}(\eta, \dot{\eta}) = \left(\dot{\mathbb{J}} - \frac{1}{2}\frac{\partial}{\partial\eta}\left(\dot{\eta}^{\mathrm{T}}\mathbb{J}\right)\right)\dot{\eta}$$
$$= C(\eta, \dot{\eta})\dot{\eta} \qquad (2.26)$$

where $C(\eta, \dot{\eta})$ is referred to as the Coriolis term and contains the gyroscopic and centrifugal terms associated with the η dependence of $\mathbb{J}$. This yields

$$m\ddot{\xi} + mg\mathbf{E}_z = \mathbf{F}_\xi \qquad (2.27)$$
$$\mathbb{J}\ddot{\eta} = \tau - C(\eta, \dot{\eta})\dot{\eta} \qquad (2.28)$$

To simplify let us take

$$\tilde{\tau} = \begin{pmatrix} \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{pmatrix} = \mathbb{J}^{-1}\left(\tau - C(\eta, \dot{\eta})\dot{\eta}\right) \qquad (2.29)$$

Finally one obtains

$$m\ddot{x} = u(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) \qquad (2.30)$$
$$m\ddot{y} = u(\cos\phi\sin\theta\sin\psi - \cos\psi\sin\phi) \qquad (2.31)$$
$$m\ddot{z} = u\cos\theta\cos\phi - mg \qquad (2.32)$$
$$\ddot{\psi} = \tilde{\tau}_\psi \qquad (2.33)$$
$$\ddot{\theta} = \tilde{\tau}_\theta \qquad (2.34)$$
$$\ddot{\phi} = \tilde{\tau}_\phi \qquad (2.35)$$

where x and y are coordinates in the horizontal plane, z is the vertical position, and $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$ and $\tilde{\tau}_\phi$ are the yawing moment, pitching moment and rolling moment, respectively, which are related to the generalized torques τ_ψ, τ_θ, τ_ϕ.

Fig. 2.4 The quad-rotor in an inertial frame. f_i represent the thrust of motor i and T_f is the main thrust
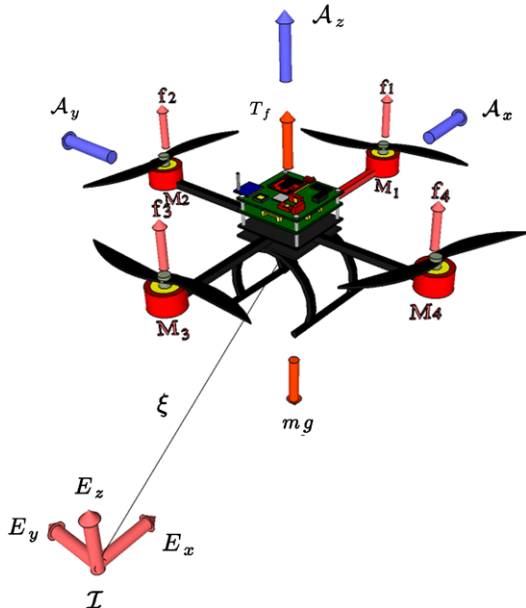
2.2.2 Newton–Euler Approach

The general motion of a rigid body in space is a combination of translational and rotational motions. Consider a rigid body moving in inertial space, undergoing both rotations and translations. Let us define now an earth fixed frame $\mathscr{I}$ and a body-fixed frame $\mathscr{A}$, as seen in Fig. 2.4. The center of mass and the body-fixed frame are assumed to coincide. Using Euler angles parametrization, the airframe orientation in space is given by a rotation R from $\mathscr{A}$ to $\mathscr{I}$, where $R \in SO(3)$ is the rotation matrix. Using the Newton–Euler formalism, the dynamics of a rigid body under external forces applied to the center of mass and expressed on earth fixed frame is

$$\begin{aligned}
\dot{\boldsymbol{\xi}} &= \mathbf{v} \\
m\dot{\mathbf{v}} &= \mathbf{f} \\
\dot{R} &= R\hat{\boldsymbol{\Omega}} \\
I\dot{\boldsymbol{\Omega}} &= -\boldsymbol{\Omega} \times I\boldsymbol{\Omega} + \boldsymbol{\tau}
\end{aligned} \tag{2.36}$$

where $\boldsymbol{\xi} = (x, y, z)^{\mathrm{T}}$ denotes the position of the center of mass of the airframe with respect to the frame $\mathscr{I}$ relative to a fixed origin, $\mathbf{v} \in \mathscr{I}$ denotes the linear velocity expressed in the inertial frame, and $\boldsymbol{\Omega} \in \mathscr{A}$ denotes the angular velocity of the airframe expressed in the body-fixed frame. The mass of the rigid body is denoted by m, and $I \in \mathbb{R}^{3 \times 3}$ denotes the constant inertia matrix around the center of mass (expressed in the body-fixed frame $\mathscr{A}$). $\hat{\boldsymbol{\omega}}$ denotes the skew-symmetric matrix of the vector $\boldsymbol{\omega}$. $\mathbf{f} \in \mathscr{I}$ represents the vector of the principal non-conservative forces applied to the object; including thrusts T_f and drag terms associated with the rotors.

$\boldsymbol{\tau} \in \mathscr{A}$ is derived from differential thrust associated with pairs of rotors along with aerodynamics effects and gyroscopic effects.

Translational Force and Gravitational Force   The only forces acting on the body are given by the translational force T_f and the gravitational force g. From Fig. 2.4, the thrust applied to the vehicle is

$$T_f = \sum_{i=1}^{4} f_i \tag{2.37}$$

where the lift f_i generated by a rotor in free air can be modeled as $f_i k \omega_i^2$ in the z-direction, where $k > 0$ is a constant and ω_i is the angular speed of the ith motor. Equation (2.37) can be rewritten as

$$T_f = k \left(\sum_{i=1}^{4} \omega_i^2 \right) \tag{2.38}$$

Then

$$\mathbf{F} = \begin{bmatrix} 0 \\ 0 \\ T_f \end{bmatrix} \tag{2.39}$$

The gravitational force applied to the vehicle is

$$\mathbf{f}_g = -mg\mathbf{E}_z \tag{2.40}$$

This yields

$$\mathbf{f} = \mathbf{R}_{E_z} T_f + \mathbf{f}_g \tag{2.41}$$

Torques   Due to the rigid rotor constraint, the dynamics of each rotor disc around its axis of rotation can be treated as a decoupled system in the generalized variable ω_i, denoting angular velocity or a rotor around its axis. The torque exerted by each electrical motor is denoted by τ_{M_i}. The motor's torque is opposed by an aerodynamic drag $\tau_{\mathrm{drag}} = k_\tau \omega_i^2$. Using Newton's second law one has

$$I_M \dot{\omega}_i = -\tau_{\mathrm{drag}} + \tau_{M_i} \tag{2.42}$$

where I_M is the angular moment of the ith motor and $k_\tau > 0$ is a constant for quasi-stationary maneuvers in free flight. In steady state, i.e., when $\dot{\omega}_i = 0$, the yaw torque is

$$\tau_{M_i} = k_\tau \omega_i^2 \tag{2.43}$$

The generalized torques are thus

$$\boldsymbol{\tau}_{\mathscr{A}} = \begin{bmatrix} \sum_{i=1}^{4} \tau_{M_i} \\ (f_2 - f_4)\ell \\ (f_3 - f_1)\ell \end{bmatrix} = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} \tag{2.44}$$

where ℓ represents the distance between the motors and the center of gravity. Rewriting (2.44) one has

$$\tau_\psi = k_\tau\left(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2\right) \tag{2.45}$$

$$\tau_\theta = \ell k\left(\omega_2^2 - \omega_4^2\right) \tag{2.46}$$

$$\tau_\phi = \ell k\left(\omega_3^2 - \omega_1^2\right) \tag{2.47}$$

where τ_ψ, τ_θ and τ_ϕ are the generalized torques (yawing moment, pitching moment and rolling moment, respectively). Each rotor may be thought of as a rigid disc rotating around the axis E_z in the body-fixed frame, with angular velocity ω_i. The rotor's axis of rotation is itself moving with the angular velocity of the frame. This leads to the following gyroscopic torques applied to the airframe:

$$\boldsymbol{\tau}_{G_{\mathscr{A}}} = -\sum_{i=1}^4 I_M(\boldsymbol{\omega}\times\mathbf{E}_z)\omega_i$$

$$= -(\boldsymbol{\omega}\times\mathbf{E}_z)\sum_{i=1}^4 I_M\omega_i \tag{2.48}$$

This yields

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathscr{A}} + \boldsymbol{\tau}_{G_{\mathscr{A}}} \tag{2.49}$$

Rewriting (2.36), one has

$$\dot{\boldsymbol{\xi}} = \mathbf{v}$$
$$m\dot{\mathbf{v}} = \mathbf{R}_{E_z}T_f - mg\mathbf{E}_z$$
$$\dot{R} = R\hat{\boldsymbol{\Omega}} \tag{2.50}$$
$$I\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega}\times I\boldsymbol{\Omega} + \boldsymbol{\tau}_{\mathscr{A}} + \boldsymbol{\tau}_{G_{\mathscr{A}}}$$

2.2.3 Newton's Equations to Lagrange's Equations

Using the classical *yaw*, *pitch* and *roll* Euler angles (ψ, θ, ϕ) applied in aeronautical applications [5, 33], the rotation matrix can be expressed as

$$R = \begin{bmatrix} c_\theta c_\psi & c_\psi s_\theta s_\phi - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\theta s_\phi s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{2.51}$$

The equations in (2.50) can be separated into the $\boldsymbol{\xi}$ coordinates dynamics and the $\boldsymbol{\eta}$ dynamics. Rewriting the $\boldsymbol{\xi}$ dynamics one has

$$\ddot{\boldsymbol{\xi}} = \frac{1}{m}(\mathbf{R}_{E_z}T_f - g\mathbf{E}_z) \tag{2.52}$$

where

$$\mathbf{R}_{E_z} = \begin{bmatrix} s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\phi s_\theta s_\psi - c_\psi s_\phi \\ c_\theta c_\phi \end{bmatrix}$$

From Figs. 2.3 and 2.4 one has $u = T_f$, this yields

$$\ddot{x} = \frac{1}{m}u(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta) \tag{2.53}$$

$$\ddot{y} = \frac{1}{m}u(\cos\phi\sin\theta\sin\psi - \cos\psi\sin\phi) \tag{2.54}$$

$$\ddot{z} = \frac{1}{m}u\cos\theta\cos\phi - g \tag{2.55}$$

From Newton–Euler formalism, one obtains in (2.53)–(2.55) the same equations as obtained in (2.30)–(2.32).

2.2.4 Newton–Euler Approach for an X-type Quad-Rotor

The quad-rotor model presented in Sects. 2.2.1 and 2.2.2 considers front and rear motors aligned with the longitudinal axis, and left and right motors aligned with the lateral axis. This section introduces an "X-type" quad-rotor flying configuration, considering two frontal motors and two rear motors. The quad-rotor dynamical model equations are based on Newton–Euler formalism, where the nonlinear dynamics is obtained in North-East-Down (NED) inertial and body-fixed coordinates, see Fig. 2.5. Let $\{N, E, D\}$ represent the inertial reference frame and $\{X, Y, Z\}$ represent the body-fixed frame. The position vector of the center of mass of the rotorcraft is denoted by $\boldsymbol{\xi} = (x, y, z)^T$, representing the position coordinates of the vehicle relative to the NED inertial frame. The orientation vector of the aircraft with respect to the inertial frame is expressed by $\boldsymbol{\eta} = (\psi, \theta, \phi)^T$, where ψ, θ and ϕ are the yaw, pitch and roll Euler angles, respectively. The full nonlinear dynamics of the quad-rotor can be expressed as

$$m\ddot{\boldsymbol{\xi}} = -mg\mathbf{D} + \mathbf{R}F \tag{2.56}$$

$$I\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega} \times I\boldsymbol{\Omega} + \boldsymbol{\tau} \tag{2.57}$$

where $R \in SO(3)$ is a rotation matrix that associates the inertial frame with the body-fixed frame, F denotes the total force applied to the vehicle, m is the total mass, g denotes the gravitational constant, $\boldsymbol{\Omega}$ represents the angular velocity of the vehicle expressed in the body-fixed frame, I describes the inertia matrix, and $\boldsymbol{\tau}$ is the total torque.

Let $u = \sum_{i=1}^{4} T_i$ be the force applied to the vehicle, which is generated by the four rotors. Assuming that this force has only one component in the Z direction, the total force can be written as $\mathbf{F} = (0, 0, -u)^T$. The rotation matrix R is defined as

$$R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{2.58}$$

Fig. 2.5 NED diagram of the quad-rotor dynamical model
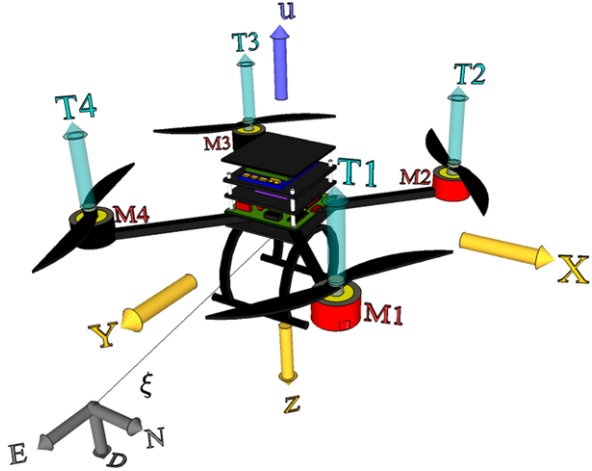
where $c_. = \cos(.)$ and $s_. = \sin(.)$. Let us define now an auxiliary vector $\tilde{\tau}$ related to the generalized torque τ and based on (2.57):

$$\tilde{\tau} = \begin{bmatrix} \tilde{\tau}_\psi \\ \tilde{\tau}_\theta \\ \tilde{\tau}_\phi \end{bmatrix} = I^{-1}W^{-1}(-I\dot{W}\dot{\eta} - W\dot{\eta} \times IW\dot{\eta} + \tau) \tag{2.59}$$

where $\boldsymbol{\Omega} = W\dot{\eta}$ and W is [38]:

$$W = \begin{bmatrix} -\sin(\theta) & 0 & 1 \\ \cos(\theta)\sin(\phi) & \cos(\phi) & 0 \\ \cos(\theta)\cos(\phi) & -\sin(\phi) & 0 \end{bmatrix} \tag{2.60}$$

Using (2.56)–(2.59), the quad-rotor dynamical model can be represented by

$$m\ddot{x} = -u\big(\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi)\big) \tag{2.61}$$

$$m\ddot{y} = -u\big(\sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi)\big) \tag{2.62}$$

$$m\ddot{z} = -u\big(\cos(\theta)\cos(\phi)\big) + mg \tag{2.63}$$

$$\ddot{\psi} = \tilde{\tau}_\psi \tag{2.64}$$

$$\ddot{\theta} = \tilde{\tau}_\theta \tag{2.65}$$

$$\ddot{\phi} = \tilde{\tau}_\phi \tag{2.66}$$

In the "X-type" quad-rotor model, the motors M_1 and M_3 rotate clockwise, while motors M_2 and M_4 rotate counter-clockwise. Assuming that total thrust approximately counteracts gravity, i.e., the quad-rotor is in hover or near-hover flight conditions, we can consider that each thrust can be modeled as $\tau_i = Cw_i^2$, where C is a constant value depending on the rotor characteristics and w_i denotes the speed of the rotor i [6]. For simplicity, it is also assumed that the torque τ_i generated by

each rotor is proportional to its lift force, then $\tau_i = C_M T_i$. Taking into account the previous assumptions, we can obtain the generalized torques as

$$
\begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} -C_M & C_M & -C_M & C_M \\ -l & -l & l & l \\ -l & l & l & -l \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{2.67}
$$

where l represents the distance between the center of mass and the center of the rotor.

2.3 Concluding Remarks

In this chapter Euler–Lagrange and Newton–Euler approaches have been applied for obtaining a simplified model of a quad-rotor rotorcraft. The model is formed by a minimum number of states and inputs, but retains the main features that must be considered when designing control laws. Two quad-rotor configurations were analyzed. The first configuration addressed a classical motor arrangement having one pair of motors aligned with the longitudinal axis while the other pair is aligned with the translational axis. The second configuration addressed an "X-Flyer" motor arrangement, having two frontal motors and two rear motors.

The models obtained here will be used in later sections for designing control laws devoted to attitude stabilization and autonomous positioning.

Chapter 3
The Quad-Rotor Experimental Platform

The quad-rotor UAV system has become a widely used platform, for this reason, many research teams and manufacturers in the world are working with this platform. Concerning commercially available quad-rotors, they are commonly provided having closed technology, so, they do not allow the inclusion of novel control laws or vision algorithms developed by the user. Thus we decide to develop a quad-rotor platform which has programs and certain electronics that were developed at HEUDIASYC laboratory (France).

Evaluating control strategies under real flight conditions requires the construction of an experimental system that must be properly designed. The developed system must be capable of estimating the variables describing the states of the vehicle, basically angular and translational position and velocity. In addition, the system must be capable of generating the control input to control such states.

This chapter is divided as follows. Section 3.1 gives general details concerning the most common sensing technologies available on UAVs. Section 3.2 presents the architecture of the systems forming the UAV platform. A supervisory ground station developed specifically for the developed systems is presented in Sect. 3.3. Section 3.4 describes the quad-rotors conceived during the research activities. With the purpose of stabilizing the quad-rotor during real-time experiments, a hierarchical control strategy is introduced in Sect. 3.5. Real-time experiments consisting on autonomous hover flights are shown in Sect. 3.6, demonstrating the vehicle's performance. Some concluding remarks are discussed in Sect. 3.7.

3.1 General Overview of UAV Sensing Technologies

Choosing adequate sensors plays a major role when designing autonomous robots. Without a reliable and fast sensing system, a good stabilization would be difficult to obtain. In order to perform autonomously, aerial vehicles must have accurate information concerning its states, as well as of the environment surrounding them. A robotic helicopter must contain sensors for measuring its angular behavior in

order to stabilize its attitude. Sensing systems for localization and environment perception are also important for achieving fully autonomy. The characteristics and number of sensors installed onboard a UAV should be chosen according to the tasks of the vehicle, but also to fit payload limitations, energy consumption and even to keep a low price of the platform.

UAVs attitude sensing is mainly performed by means of inertial sensors, some of which are listed next.

- **Gyroscope:** it measures the angular velocity of a system in the inertial reference frame. Microchip-packaged MicroElectroMechanical Systems (MEMS) gyroscopes are commonly used for the stabilization of UAVs.
- **Accelerometer:** it measures the linear acceleration of a system in the inertial reference frame. Modern accelerometers are often small MEMS, consisting of a cantilever beam with a proof mass. They can be used to measure inclination, dynamic distance and speed with or without the influence of gravity.
- **Magnetometer/Compass:** device used for measuring the magnetic field of the Earth to ascertain the North Pole direction. Normally, a magnetometer provides the yaw angle of the aircraft.
- **Inertial Measurement Unit (IMU):** electronic device that measures and reports on a craft's velocity, orientation, and gravitational forces, using a combination of accelerometers, gyroscopes and a compass. IMUs are used to maneuver aircrafts (including UAVs), spacecrafts and satellites. Commonly, the IMU is the main component of the navigation systems. The data collected from the IMU's sensors allow a computer to track the vehicle's position, using a method known as dead reckoning.

The most common sensors for localization and environment perception are

- **Global Positioning System (GPS):** a space-based global navigation satellite system (GNSS) that provides reliable absolute location anywhere on Earth, when there is an unobstructed line of sight to four or more GPS satellites. GPS report aircraft-specific information such as speed, bearing and altitude.
- **Laser Range Finder (LRF):** device which uses a laser beam to determine the distance to an object. LRFs operate on the time of flight principle by sending a laser pulse in a narrow beam towards the object and measuring the time taken by the pulse to be reflected off the target and returned to the sender. LRFs are used extensively in 3D object recognition and modeling and in a wide variety of computer vision-related fields, offering high-precision scanning abilities, with either single-face or 360-degree scanning modes.
- **Ultrasonic Sensor:** it generates high frequency sound waves and evaluates the echo which is received back by the sensor. To determine the distance to an object, the sensor calculate the time interval between sending the signal and receiving the echo.
- **Infrared Sensor:** it transmits infrared pulses and measures the time it takes to return back, estimating in this way a distance. It works for shorter distances than the ultrasonic sensors.

- **Pressure Sensor:** it generates a signal as a function of the pressure imposed. In aircraft, rockets, satellites and weather balloons, pressure sensors are used for altitude sensing, using the relationship between changes in pressure relative to the altitude.
- **Imaging Sensors:** visual sensing is especially attractive because it is passive, non-contact, very versatile and low-cost. It provides a tremendous amount of information about a robot's environment, and it is potentially the most powerful source of information among all the sensors used on robots to date. However, it introduces several technical challenges: the mapping from the image plane to 3-dimensional coordinates can be significantly nonlinear. Image processing is computationally intensive and introduces (typically variable) latencies from the time of capture to the time measurements are available. Occlusions, poor lighting, or failure of the image processing algorithms can result in loss of measurements for long periods of time.

In an ideal situation, the information provided by the previously listed sensors should be treated onboard the UAV. However, some of the sensors, mainly imaging sensors, provide data that must be treated intensively in order to obtain the required information. This intensive processing will certainly overload the embedded processor that stabilizes the aircraft attitude. Such situation motivated the development of robotics systems capable of communicating with a supervisory ground station placed in a remote location, with enough computational power to perform image processing in an appropriate amount of time.

The challenge now extends to the development of a system consisting of:

- A UAV equipped with a sensor suit well adapted to the vehicle tasks.
- A remote supervisory ground station capable of processing large amount of information in real time.
- A wireless link between the UAV and the ground station. The UAV sends images to the ground station responsible of performing all the intensive image processing tasks. Finally, the ground station sends back the data required for the UAV stabilization.

The architecture of a system with such characteristics, which has been designed to fit the research needs of this book, is presented in the next section.

3.2 System Architecture

In order to maintain control over the quad-rotor, the onboard system must perform quickly and reliably. At the same time, the supervisory ground station must provide useful information to the onboard system even when such data are asynchronous or unreliable. Communication plays a major role between both systems, as well as between the group of programs running in the ground station computer.

The proposed configuration consists of a quad-rotor helicopter and a ground station PC, communicated by wireless data and video links. Such architecture is shown
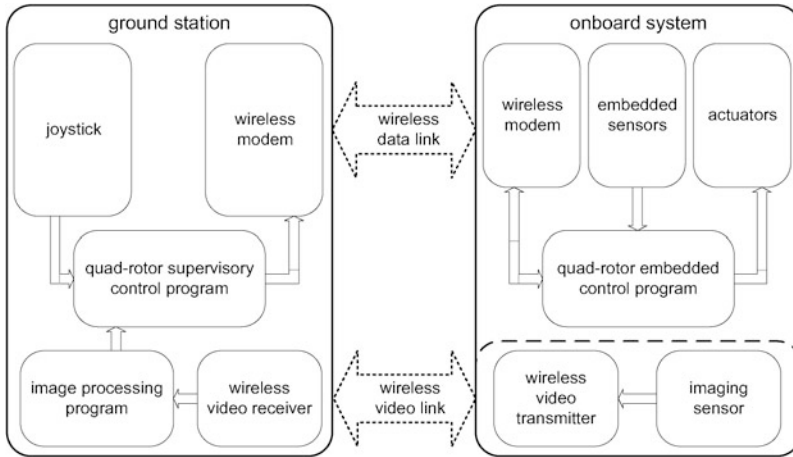
Fig. 3.1 System architecture

in Fig. 3.1. The systems onboard the quad-rotor includes the embedded processor, the components for sensing and propulsion, the data wireless link and an imaging sensor connected to a device capable of wireless video transmission. The off-board system consists of a ground station PC connected to a flight simulator Joystick, wireless data link and a wireless system for receiving video. The joystick enables a human pilot to manually control the aerial vehicle, sending control commands by means of the wireless data link. The ground station runs programs for retrieving information concerning the helicopter status, as well as image processing programs for information extraction.

3.3 Supervisory Ground Station

The supervisory ground station consists of a personal computer, a flight simulator joystick and two wireless links, one for data transmission and the other for receiving video signal. This architecture is shown in the left side of Fig. 3.1.

The joystick, see Fig. 3.2(a), is connected to the computer via a USB 2.0 port. The joystick input data are continuously checked by a supervisory control application, which communicates periodically with the quad-rotor using a Maxstream Xbee-PRO Zigbee modem (wireless data link working in the 2.45 GHz band). This wireless link, shown in Fig. 3.2(b), is also used to receive and save information sent periodically by the aerial vehicle, which helps to characterize its performance during experiments. The wireless video link provides real-time video coming from an imaging system onboard the UAV. The ground station runs a computer vision program developed for performing image processing and information extraction. The information extracted by those algorithms is placed on a fixed memory segment that is shared with the supervisory control application. With this method, the supervisory control application can also take control of the vision system.

flight
models

throttle

(a)   (b)
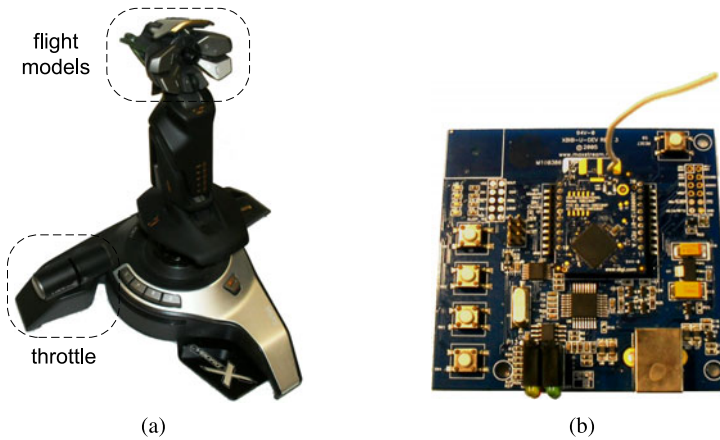
Fig. 3.2 Supervisory ground station components: (**a**) flight simulator joystick; (**b**) Maxstream Xbee-PRO Zigbee

Table 3.1 Information packages description

Package	Start	Data	Checksum	Total
Sent bytes	3	10	2	15
Received bytes	3	25	2	30

The main function of the base station is the supervisory control application, since it coordinates the running programs and allows to piloting the quad-rotor. Within the supervisory application, different UAV flight modes can be chosen: manual control, altitude stabilization by means of the altitude sensors, vision-based position stabilization and navigation. A condition for turning on the motors and an emergency stop signal are also available. The previous features are selected by the user via the joystick buttons.

Data transmission between the supervisory ground station and the aerial vehicle is performed at a frequency of 30 Hz. The data sent from the base station to the UAV are prioritized since they carry information concerning position control. The other way around, the quad-rotor sends its data only after successfully having received a data packet from the base station. The data sent to the aerial vehicle contain first three bytes allowing the robot to recognize the type of command, next, ten data bytes with the joystick information and the flight modes command, and finally two bytes for checksum test, allowing a package validation. The data received from the quad-rotor contain first the same three bytes, followed by the data bytes containing sensor data and control information, and finally the packet ends with a checksum for package validation. The information packages sent and received by the supervisory station are described in Table 3.1.

3.4 Quad-Rotor Design

A quad-rotor helicopter is composed basically of a four-arms frame carrying the next components:

- A propulsion system.
- A sensing system.
- Wireless communication links.
- Embedded processors for
  - Reading sensors data.
  - Manage communications.
  - Control law computation.
- Energy supply.

The objective when designing a quad-rotor is to develop a vehicle capable of robust autonomous hover, with a minimum position drift, and well suited to the implementation of control strategies for positioning and navigation.

A UAV needs to know accurate information about itself and its surrounding environment, therefore, onboard sensors play a major role in their design. As mentioned previously, the group of sensors depends on the vehicle's tasks, the environment where it performs, the payload limitations, computation power, energy requirements and cost. For a quad-rotor vehicle, critical sensors are those measuring its angular positions and angular rates. Such variables can be measured by a group of gyroscopes and accelerometers, or by an IMU. This group of inertial sensors provides the required data for the attitude controller. Knowing the vehicle's position, translational velocity and acceleration is also of great importance for positioning control and autonomous navigation. Absolute positioning with respect to world coordinates can be obtained from a GPS, however, this system cannot provide a relative position of the vehicle with respect to a target (for a landing maneuver for example) or an obstacle (for performing obstacle avoidance). Furthermore, GPS signals are only available outdoors and in environments free of signal blockages. Such disadvantages led to the decision of using alternative choices for estimating the quad-rotor's 3-dimensional position.

Since the present research focuses on developing an aerial vehicle capable of performing indoors as well as outdoors, relative positioning sensors seem to be more appropriated. After an evaluation of the existing technologies, it was decided to equip the quad-rotor with an sensor suit consisting of an IMU and three analog rate gyros for measuring Euler angles and angular rates, respectively, an ultrasonic sensor and an atmospheric pressure sensor for altitude measurements, as well as of an imaging system for computing relative position and translational velocity. During research activities, the quad-rotor prototype went through a constant evolution. Modifications were mainly performed for increasing the vehicle's sensory capacities, but also for increasing its autonomy and payload. Three fully functional quad-rotor helicopters emerged from this research. The three of them are based on the Texas Instruments® TMS320F2812 digital signal processor (DSP) [78], whose main characteristics are shown in Table 3.2.

Table 3.2 Characteristics of the TI TMS320F2812 DSP

TI TMS320F2812 DSP	
CPU	32-bit
Clock	150 MHz
Power	1.9 V @ 150 MHz
Flash memory	128 K
Serial ports	2
ADC channels	16 @ 12-bit
PWM channels	12 @ 16-bit

The TI DSP counts with two RS232 serial ports. One of them deals with the wireless communication between the quad-rotor and the supervisory ground station, which is performed at 30 Hz, while the second port reads IMU measurements at 100 Hz. Analog to digital converters (ADC) are used to read three analog gyros and three accelerometers, one ultra sonic sensor and one pressure sensor, four current monitors and one battery level monitor. ADCs readings are performed at a maximum rate of 1 KHz and a minimum of 500 Hz. Using the signals previously explained, the DSP computes the control input that stabilizes the helicopter during flight, which is sent to the motor's electronic speed control (ESC) in the form of pulse-width modulation (PWM) signals at 500 Hz. The design specifications that have been implemented successfully in the embedded DSP are shown in Table 3.3. The next subsections gives details concerning the specific architecture of each one of the three quad-rotors developed.

3.4.1 Cross-Flyer Design

The structural frame of the "Cross-Flyer" Design consists of a 40 cm MK40 frame-set from MikroKopter® [61]. It is arranged in a conventional form, having the

Table 3.3 Design specifications of the TI TMS320F2812 DSP

Specification	Device	Interface	Processing rate
Ground station comm.	Wireless modem	RS232	30 Hz
IMU reading	IMU	RS232	100 Hz
Angular rates	Gyroscope	ADC	up to 1 KHz
Acceleration	Accelerometer	ADC	up to 1 KHz
Altitude < 6 m	Ultra Sonic	ADC	up to 1 KHz
Altitude > 6 m	Pressure sensor	ADC	up to 1 KHz
Motor current	Shunt + Op. Amp	ADC	up to 1 KHz
Battery level	Op. Amp	ADC	up to 1 KHz
Motor speed	ESC	PWM	500 Hz

front–rear motors aligned with the longitudinal axis, while the left–right motors
are aligned with the translational axis. The motor chosen for this quad-rotor is the
Booster 1200, with a maximum rotation speed of 1200 rpm and providing a lift of
350 grams each. The propeller installed to each motor is made of plastic, measuring
25 cm long. Each motor-propeller unit has a weight of 40 grams. The motor's ESC
is the PM-30A from Pro-Tronik. This device receive the speed command in form of
PWM at 500 Hz, producing a three phase AC at a maximum rate of 200 Hz. Finally,
a 12 V–2200 mAh Li-Po battery from Pro-Tronik provides power supply to all the
electronic system onboard. Nuts and screws made of plastic were used for fixing
components whenever possible, with the purpose of keeping the vehicle weight to a
minimum.

The embedded electronics of the "Cross Flayer" is composed of two intercon-
nected cards: the first board is a control unit, while the second one deals with the
propulsion system and motor current monitoring. The control unit card performs the
essential tasks of sensing, communicating and stabilizing the UAV attitude during
fly. Its properties can be summarized as follows.

- **Processor:** Texas Instruments® TMS320F2812 DSP.
- **Inertial sensors:**
  - An MIDG II INS/GPS IMU from Microbotics Inc® measures the angular po-
    sition of the rotorcraft.
  - Three ADXRS150 analog gyroscopes measure the angular rates. Analog rate
    measurement are preferred over IMU-based measurements, since a faster re-
    fresh of angular rates enables a better attitude stabilization of the UAV.
- **Atmospheric pressure sensor:** a Freescale® MPXH4115A pressure sensor, in
  combination with an appropriate amplifier circuit, measures the altitude of the
  engine for heights over 6 m.
- **Wireless link:** a XBee ZB ZigBee PRO® radio modem is used to link the su-
  pervisory ground station and the quad-rotor. This communication link introduces
  external control input and sends the sensors information to the ground station.

The propulsion system board contains:

- **Signal conditioning circuitry:** In this stage, the motor's control signals are de-
  coupled from the rest of the electronic systems. PWM signals are also filtered and
  conditioned.
- **Motor current monitoring:** This circuit allows the measurement of the current
  passing through each one of the motors. Current measuring is achieved using a
  shunt resistor connected to a shunt current monitor. The current monitor output
  signal is a voltage proportional to the current passing through the motor. This sig-
  nal is sent to the DSP ADCs via the boards connection bus. The objective of this
  current sensing system is the improvement of the quad-rotor stabilization [71].

Both Cross-Flyer electronic cards are shown in Fig. 3.3. The Cross-Flyer quad-rotor
prototype is presented in Fig. 3.4. This corresponds to the first platform developed
during the research activities. Some of its characteristics are resumed in Table 3.4.

(a)   (b)
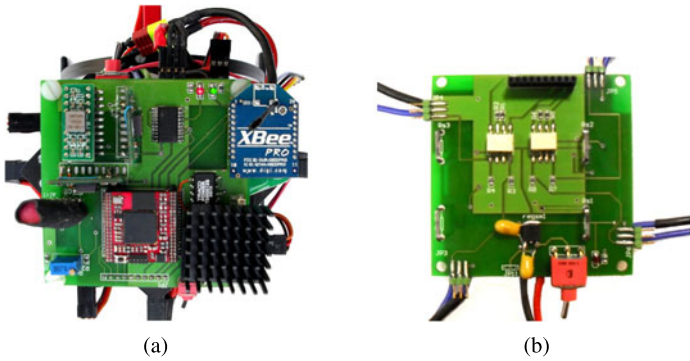
Fig. 3.3 The Cross-Flyer electronic cards: (**a**) control unit; (**b**) propulsion unit

Fig. 3.4 The Cross-Flyer
quad-rotor prototype

Table 3.4 Characteristics of
the Cross-Flyer rotorcraft

Parameter	Value
Distance between rotors	40 cm
Weight	900 grams
Payload	150 grams
Autonomy	15 minutes
Power	12 V, 2200 mAh Li-Po battery
Motor	1200 rpm brushless

3.4.2 X-Flyer Design

Commonly, quad-rotors have a pair of rotors aligned with the pitch axis, while the
other pair of rotors is aligned with the roll axis (like the Cross Flyer presented in
Sect. 3.4.1). Notice that with this configuration, the quad-rotor has only the strength

of one motor, for example the rear motor, while moving forward. The same situation stands for the lateral displacements. This analysis motivated the development of an "X-Flyer" quad-rotor having two frontal motors and two rear motors, with the purpose of improving the performance achieved with the conventional "cross" configuration. It was found experimentally (as will be seen in later sections) that the "X-Flyer" performs better when performing displacements. In addition, this configuration is more responsive while correcting induced external perturbations. Another important reason for choosing the "X-Flyer" configuration as more adequate to the research objectives is because UAVs vision systems are commonly installed pointing downwards and forwards. Consider the vision system pointing forwards for performing obstacle detection and avoidance, evidently, it is desirable to have a clear view of the scene in front of the helicopter.

In the "X-Flyer" quad-rotor the MK40-Frameset from MikroKopter, the controller–motor–propeller combination and the power supply battery used in the "Cross Flyer" quad-rotor have been retained. The embedded electronics of the "X-Flyer" is composed also of two interconnected cards: the first board is the control unit and the second board deals with the propulsion system and the current monitoring circuitry. The control unit card performs the essential tasks of sensing, communicating and stabilizing the UAV attitude during fly. These board properties can be summarized as follows.

- **Processor:** Texas Instruments® TMS320F2812 DSP.
- **Inertial sensors:**
  - A 3DM-GX1 IMU from Microstrain® is used to measure the Euler angles of the rotorcraft.
  - Three ADXRS600 analog gyroscopes for angular rates measurements.
  - An MXR9500MZ three axis accelerometer from MEMSIC® is used to estimate the helicopter linear accelerations, aiming at improving hover flight.
- **Ultrasonic sensor:** the LV-MaxSonar EZ4 ultrasonic sensor is used for measuring altitudes from 15 cm to 6 m.
- **Atmospheric pressure sensor:** a Freescale™ MPXH6115A pressure sensor is used in combination with an appropriate amplifier circuit to measure the altitude of the vehicle (for altitude > 6 m). The MPXH6115A shows good performance with respect to temperature variations that generally affects the altitude measurements.
- **Battery voltage measurement circuit:** circuitry providing the tension level of the supply battery. This information is used for several goals: perform a safety landing and turn-off before an unwanted lack of tension (avoiding accidents). Also, the supply voltage measurement is used in a processing stage of the atmospheric pressure sensor measurements.
- **Wireless link:** an XBee ZB ZigBee PRO® radio modem links the supervisory ground station and the aerial vehicle. This communication link enables to introduce external control input to the vehicle, as well as sending the helicopter sensors data to the ground station.

The propulsion system board contains:

(a)   (b)
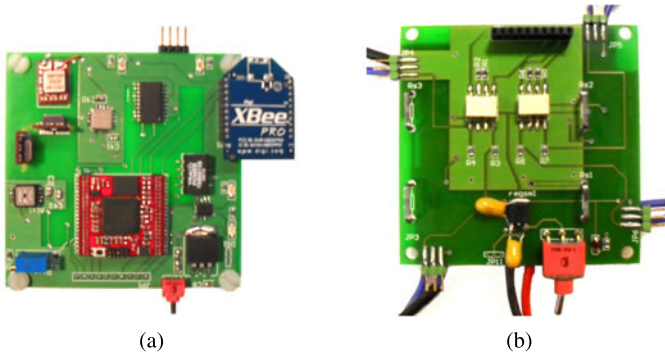
Fig. 3.5 The X-Flyer electronic cards: (**a**) control unit; (**b**) propulsion unit

Fig. 3.6 The Cross-Flyer
quad-rotor prototype

- **Signal conditioning circuitry:** In this stage, the motor's control signals are de-coupled from the rest of the electronic systems. PWM signals are also filtered and conditioned.
- **Motor current monitoring:** This circuit allows the measurement of the current passing through each one of the motors. The current measuring is achieved using a shunt resistor connected to a shunt current monitor. The current monitor output signal is a voltage proportional to the current passing through the motor. This signal is sent to the DSP ADCs via the boards connection bus. The objective of this current sensing system is the improvement of the quad-rotor stabilization [71].

Both "X-Flyer" cards are shown in Fig. 3.5. The "X-Flyer" quad-rotor prototype is presented in Fig. 3.6. Some of its characteristics are resumed in Table 3.5.

3.4.3 Improved X-Flyer Design

From experiments conducted using the previous platform, it became evident that the "X-Flyer" quad-rotor suited most of the requirements of the research objective. However, when trying to increase the number of sensors onboard, the payload

Table 3.5 Characteristics of
the X-Flyer rotorcraft

Parameter	Value
Distance between rotors	40 cm
Weight	900 grams
Payload	150 grams
Autonomy	15 minutes
Power	12 V, 2200 mAh Li-Po battery
Motor	1200 rpm brushless
Safety	voltage measurement

limitations of the "X-Flyer" came to light. Actions taken to overcome this barrier
consisted basically on installing a new set of motors with higher thrust capacity. In
addition, the electronic onboard was modified, replacing the pair of cards with and
improved single board.

The "X-Flyer" body frame is the MK-40 from MikroKopter. The motors chosen
for this version are the BL-Outrunner 2827-35 from Robbe ROXXY, providing a
maximum thrust of 820 grams each. Motors are connected to PM30A ESC from
Pro-Tronik. It has been found experimentally that this driver-motor combination
is a very effective arrangement considering energy consumption and payload lift.
Energy is provided by a 12 V–2200 mAh Li–Po battery from Pro-Tronik.

The embedded electronics of the improved X-Flyer consists of a single card,
where the next group of components are installed.

- **Processor:** Texas Instruments® TMS320F2812 DSP.
- **Inertial sensors:**
  - An MIDG II INS/GPS IMU from Microbotics Inc® measures the angular position of the rotorcraft.
  - Three ADXRS610 MEMS analog rate sensors measure angular rates.
  - An MXR9500MZ three axis accelerometer from MEMSIC® estimates linear accelerations, aiming at improving the helicopter hover flight performance.
- **Ultrasonic sensor:** the LV-MaxSonar EZ4 ultrasonic sensor is used for measuring altitudes from 15 cm to 6 m.
- **Atmospheric pressure sensor:** a Freescale™ MPXH6115A pressure sensor is installed to measure the flight altitude of the quad-rotor (for altitude > 6 m).
- **Battery voltage measurement circuit:** a circuit to measure battery voltage is also available, which helps to determine when the power supply is going under the operational values. The supply battery level is also used in a preprocessing stage of the measurements from the atmospheric pressure sensor.
- **Wireless link:** an XBEE09P radio modem from Digi International Inc. enables a wireless communication between the quad-rotor and a remote ground station PC. It is used to receive UAV data in real time, and also to send control commands for stabilizing the helicopter.
- **Signal Conditioner:** each control input of the four motors is decoupled from the rest of the electronic systems. The PWM signals are also filtered and conditioned.

Fig. 3.7 Improved X-Flyer electronic board design
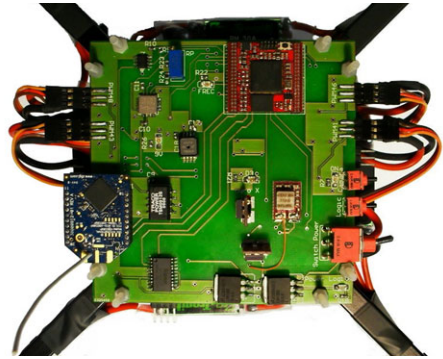
Fig. 3.8 The Improved X-Flyer quad-rotor platform

Table 3.6 Characteristics of the Improved X-Flyer rotorcraft

Parameter	Value
Distance between rotors	40 cm
Weight	900 grams
Payload	600 grams
Autonomy	15 minutes
Power	12 V, Li-Po battery
Motor	5850 rpm brushless
Safety	Voltage measurement

The Improved X-Flyer electronic card is shown in Fig. 3.7. The robotic helicopter as described is shown in Fig. 3.8. It has a total weight of 945 grams. The maximum time of flight autonomy achieved is 15 minutes. It has a payload capacity of 600 grams, which considerably increases the number of sensors that can be installed onboard. Some of its characteristics are resumed in Table 3.6.

3.5 Hierarchical Control Strategy

A cascaded control system allows to separate the position control from the attitude control, not only in theoretical terms, but also in the hardware design of the quad-rotor. Commonly a wireless communication is established between an aerial robot and its corresponding supervisory ground station. The cascaded controller allows to take advantage of this wireless link to perform deported position control tasks, which can be generated by a user using a joystick, or from a secondary application running in the supervisory station. The control scheme for the overall cascaded system is then formed by a position controller and an attitude controller, where the position controller generates the required and desired angles to the attitude controller.

The cascaded control for the quad-rotor is implemented adopting a hierarchical control strategy. Considering the time-scale separation between the positioning control and the attitude dynamics, the dynamical model can be seen as a series of nested subsystems. The first two subsystems to be stabilized are the altitude and the yaw dynamics. In this approach, the position control represents a high-level control running off-board, while the attitude control is a low-level control performed by the autopilot embedded in the vehicle.

3.5.1 Altitude and Yaw Control

The control of the vertical position expressed in (2.32) can be achieved by using the following control input:

$$u = (r_1 + mg) \frac{1}{\cos\theta \cos\phi} \tag{3.1}$$

where

$$r_1 = -k_{vz}\dot{z} - k_{pz}e_z \tag{3.2}$$

with $e_z = z - z_d$ as the z error position and z_d as the desired altitude. k_{pz} and k_{vz} are positive constants. Thus, for the altitude dynamics, r_1 is a PD controller. In the case of the yaw angular position in (2.33) one uses

$$\tilde{\tau}_\psi = -k_{v\psi}\dot{\psi} - k_{p\psi}e_\psi \tag{3.3}$$

where $e_\psi = \psi - \psi_d$ denotes the yaw error, ψ_d represents the desired yaw angle, $k_{p\psi}$ and $k_{v\psi}$ denote the positive constants of a PD controller. Indeed, introducing (3.1) and (3.3) into the set of equations (2.30)–(2.35) and provided that $\cos\theta \cos\phi \neq 0$, one obtains

$$m\ddot{x} = (r_1 + mg)\left(\frac{\sin\psi \tan\phi}{\cos\theta} + \cos\psi \tan\theta\right) \tag{3.4}$$

$$m\ddot{y} = (r_1 + mg)\left(\sin\psi \tan\theta - \frac{\cos\psi \tan\phi}{\cos\theta}\right) \tag{3.5}$$

$$m\ddot{z} = -k_{vz}\dot{z} - k_{pz}e_z \tag{3.6}$$

$$\ddot{\psi} = -k_{v\psi}\dot{\psi} - k_{p\psi}e_\psi \tag{3.7}$$

The control parameters $k_{p\psi}$, $k_{v\psi}$, k_{pz} and k_{vz} should be carefully chosen to ensure a stable well-damped response in the vertical and yaw axes [27]. From (3.6) and (3.7) it follows that $\psi \to \psi_d$ and $z \to z_d$. Note that from (3.2)–(3.3) and (3.6)–(3.7) $r_1 \to 0$ and $\psi \to \psi_d$. For a time T large enough, e_z and e_ψ are arbitrarily small, therefore, (3.4) and (3.5) reduce to

$$\ddot{x} = g\tan\theta \tag{3.8}$$

$$\ddot{y} = -g\frac{\tan\phi}{\cos\theta} \tag{3.9}$$

3.5.2 Control of Forward Position and Pitch Angle

Consider the subsystem given by (2.34) and (3.8). To further simplify the analysis, impose a very small upper bound on $|\theta|$ in such a way that the difference $\tan(\theta) - \theta$ is arbitrarily small ($\theta \approx \tan(\theta)$). Therefore, the subsystem (2.34) and (3.8) becomes the following linearized system:

$$\ddot{x} = g\theta \tag{3.10}$$

$$\ddot{\theta} = \tilde{\tau}_\theta \tag{3.11}$$

The longitudinal subsystem focuses on the $x - \theta$ stabilization. Indeed, the goal is to control the pitch θ and the longitudinal position x to a desired references. Consider the linearized system (3.10)–(3.11), the longitudinal position of the vehicle can be described as a chain of integrators

$$\dot{x}_1 = x_2 \tag{3.12}$$

$$\dot{x}_2 = x_3 \tag{3.13}$$

$$\dot{x}_3 = x_4 \tag{3.14}$$

$$\dot{x}_4 = \tilde{\tau}_\theta \tag{3.15}$$

where x_1 is the error between the desired and the actual position, $x_3 = \theta$ and $x_4 = \dot{\theta}$. The hierarchical control is constructed by separating, in the previous subsystem, the rotational dynamics from the translational dynamics (low-level control and position control, respectively). Considering x_3 as the control input for the translational dynamics, and applying a backstepping change of variables, one has

$$\dot{x}_1 = x_2 \tag{3.16}$$

$$\dot{x}_2 = x_3^{\text{ref}} + \tilde{x}_3 \tag{3.17}$$

$$\dot{\tilde{x}}_3 = \tilde{x}_4 \tag{3.18}$$

$$\dot{\tilde{x}}_4 = \ddot{x}_2^{\text{ref}} + \tilde{\tau}_\theta \tag{3.19}$$

Here $x_3^{\text{ref}} = \theta_{\text{ref}}$, $\tilde{x}_3 = \theta - \theta_{\text{ref}}$, and $\tilde{x}_4 = \dot{\theta} - \dot{\theta}_{\text{ref}}$. θ_{ref} is the reference angle chosen to control the longitudinal displacement. Notice that setting the reference pitch angle to zero equals to stop the displacement. Since the low-level (rotational) control runs at highest rate than the positioning control, the reference angle dynamics can be ignored, then ($\dot{\theta}_{\text{ref}} = \ddot{\theta}_{\text{ref}} = 0$). Indeed, the dynamics of θ will converge faster than the position control. Then, one has

$$\dot{x}_1 = x_2 \tag{3.20}$$

$$\dot{x}_2 = x_3^{\text{ref}} + \tilde{x}_3 \tag{3.21}$$

$$\dot{\tilde{x}}_3 = x_4 \tag{3.22}$$

$$\dot{x}_4 = \tilde{\tau}_\theta \tag{3.23}$$

The longitudinal position is stabilized by defining the reference pitch angle as a function of the longitudinal position and velocity

$$\theta_{\text{ref}} = V_x(x, \dot{x}) \tag{3.24}$$

which can be defined as PD controller. The pitch reference angle is used in the rotational control as

$$\tilde{\tau}_\theta = -k_v^\theta x_4 - k_p^\theta \tilde{x}_3 \tag{3.25}$$

$$\tilde{\tau}_\theta = -k_v^\theta \dot{\theta} - k_p^\theta (\theta - \theta_{\text{ref}}) \tag{3.26}$$

The controller gains k_v and k_p should be chosen appropriately so that the polynomial $s^2 + k_v s + k_p$ is Hurwitz.

3.5.3 Control of Lateral Position and Roll Angle

Consider the subsystem given by (2.35) and (3.9). Impose a very small upper bound on $|\phi|$ in such a way that the difference $\tan(\phi) - \phi$ is arbitrarily small ($\phi \approx \tan(\phi)$). Therefore, the subsystem (2.35) and (3.9) becomes

$$\ddot{y} = -g\phi \tag{3.27}$$

$$\ddot{\phi} = \tilde{\tau}_\phi \tag{3.28}$$

The translational subsystem focuses on the y–ϕ stabilization. Consider the linearized system (3.27)–(3.28), the translational position of the vehicle can be described as a chain of integrators

$$\dot{y}_1 = y_2 \tag{3.29}$$

$$\dot{y}_1 = y_2 \tag{3.30}$$

$$\dot{y}_2 = y_3 \tag{3.31}$$

$$\dot{y}_3 = \tilde{\tau}_\phi \tag{3.32}$$

Considering ϕ as the control input for the translational dynamics, then the chain of four integrators can be separated into two subsystems composed of two integrators for both of them. The implementation of a backstepping change of variables yields

$$\dot{y}_1 = y_2 \tag{3.33}$$

$$\dot{y}_2 = y_3^{\text{ref}} + \tilde{y}_3 \tag{3.34}$$

$$\dot{\tilde{y}}_3 = y_4 \tag{3.35}$$

$$\dot{y}_4 = \tilde{\tau}_\phi \tag{3.36}$$

where $y_3^{\text{ref}} = \phi_{\text{ref}}$. Using high gains in the low-level attitude control, the dynamics of the reference angle can be ignored. The lateral position is then stabilized by defining the reference roll angle as a function of the lateral position and velocity

$$\phi_{\text{ref}} = V_y(y, \dot{y}) \tag{3.37}$$

which can be defined as PD controller. The roll reference angle is used in the rotational control as

$$\tilde{\tau}_\phi = -k_v^\phi y_4 - k_p^\phi \tilde{y}_3 \tag{3.38}$$

$$\tilde{\tau}_\phi = -k_v^\phi \dot{\phi} - k_p^\phi (\phi - \phi_{\text{ref}}) \tag{3.39}$$

The controller gains k_v and k_p should be chosen appropriately so that the polynomial $s^2 + k_v s + k_p$ is Hurwitz.

   The method presented allows a separation between the attitude control and the position control. Since the altitude sensor is attached to the aerial robot, the altitude control can also be separated from the lateral and forward position control. Then, the low-level control embedded on the quad-rotor deals with the computation of the following control laws:

$$u = (-k_{vz}\dot{z} - k_{pz}e_z + mg)\frac{1}{\cos\theta\cos\phi}$$

$$\tilde{\tau}_\psi = -k_{v\psi}\dot{\psi} - k_{p\psi}e_\psi$$

$$\tilde{\tau}_\theta = -k_v^\theta \dot{\theta} - k_p^\theta(\theta - \theta_{\text{ref}})$$

$$\tilde{\tau}_\phi = -k_v^\phi \dot{\phi} - k_p^\phi(\phi - \phi_{\text{ref}})$$

The overall distributed controller is shown in Fig. 3.9. It is modeled as a distributed system, conformed by a low-level controller computed in the embedded autopilot and a positioning controller which runs in the supervisory base station. This kind of control is effective if the translational dynamics is controlled in a slower rate than the rotational dynamics.

3.6 Autonomous Hover Flight Experiments

In order to evaluate the vehicle's performance, the development of each platform was followed by real-time experiments consisting of autonomous hover flights,
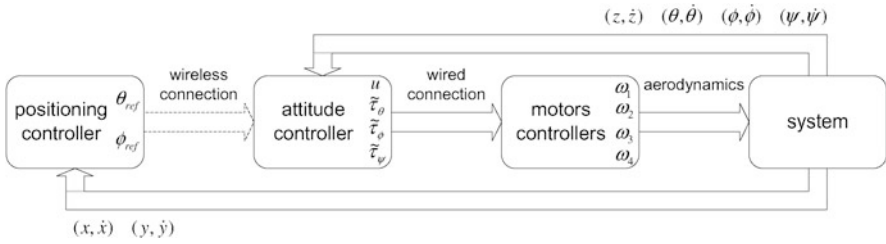
$$(z,\dot{z}) \quad (\theta,\dot{\theta}) \quad (\phi,\dot{\phi}) \quad (\psi,\dot{\psi})$$

positioning controller θ_{ref} ϕ_{ref} — wireless connection → attitude controller u $\tilde{\tau}_\theta$ $\tilde{\tau}_\phi$ $\tilde{\tau}_\psi$ — wired connection → motors controllers ω_1 ω_2 ω_3 ω_4 — aerodynamics → system

$$(x,\dot{x}) \quad (y,\dot{y})$$

**Fig. 3.9** Quad-rotor distributed control system

where the algorithm running in the embedded DSP was responsible of the aircraft attitude stabilization.

The embedded DSP manages the aircraft stabilization by using the Euler angles provided by the IMU, as well as the angular rates measured by the analog gyros. Those measurements are treated by a filtering stage in order to reduce possible noise. Next, the DSP uses the filtered values to compute the control law that stabilizes the aircraft attitude. Finally, the computed controls are sent to the corresponding motor ESC in form of PWM. The goal is to keep the pitch and roll angles near zero degrees, in order to avoid translational drift as much as possible. Figure 3.10 shows a schematic representing the algorithm running on the DSP, which is designed to take advantage of the fast interrupt response available in the DSP. The peripherals are interfaced by using interruptions, this allows the two serial communications (wireless data link and IMU measurements), and the ADC routines (gyros, ultrasonic and pressure sensors, battery level and motor's current sensing) to perform in parallel with the main control loop. Thus, the algorithm execution can be performed at rates up to 1 kHz. During experiments, the aircraft was flown at an altitude of 1.5 m approx., and was let to evolve freely. The supervisory ground station retrieved the signals describing the platform angular behavior while hovering. The performance of each one of the three vehicles is shown next.

3.6.1 Cross-Flyer Hover Graphics

The performance of the "Cross-Flyer" quad-rotor during real-time attitude stabilization is shown. Euler angles were provided by the Microbotics IMU while angular rates were measured from a set of three ADXRS150 analog rate gyroscopes. Figure 3.11(a) shows the platform Euler angles. It can be seen that the embedded control system ensures that Euler angles remain always between $(+3°, -3°)$. Similarly, angular rates measurements in Fig. 3.11(b) are relatively small. This behavior validates the effectiveness of the "Cross-Flyer" quad-rotor platform. The maximum time of flight achieved with this vehicle was 15 min, which degrades quickly as new sensors (payload) are added and also if external perturbations are induced. A video of the experiment can be seen in http://www.youtube.com/watch?v=l22PDf _WOOU.

Fig. 3.10 Schematic of the
algorithm running on the DSP
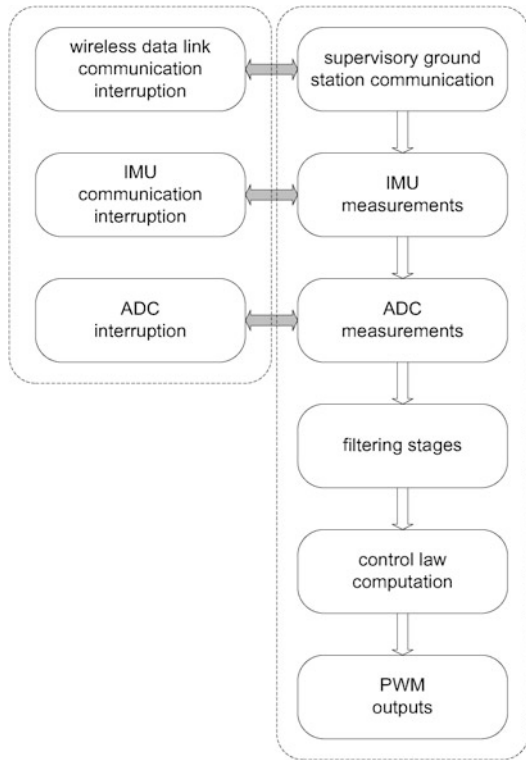
3.6.2 X-Flyer Hover Graphics

The "X-Flyer" quad-rotor performance is shown. The differences between this vehicle and the "Cross-Flyer" are the motors arrangement, the IMU used and the analog rate sensors. Using a Microstrain® IMU, this vehicles is supposed to present poorer performance. However, the ADXRS610 MEMS analog rate sensors helps to equilibrate the resulting embedded sensing system.

Figure 3.12 shows the Euler angles and angular rates experimented by the platform when performing the experiments. During this test, external perturbations in pitch and roll angles were induced in order to verify the robustness of the vehicle. Figure 3.12(a) shows abrupt changes in the pitch angle measurement at 70 s and 75 s, which are of the order of $(+10°, -10°)$. It can be seen also that after the induced perturbations, the attitude stabilization algorithm managed to correct quickly such angle variations, returning immediately the vehicle's Euler angles to values between $(+3°, -3°)$. Angular rates measurements in Fig. 3.12(b) are relatively small and clean signals.

This behavior validates the effectiveness of the "X-Flyer" quad-rotor platform with its motor arrangement. The maximum time of flight achieved with this vehicle
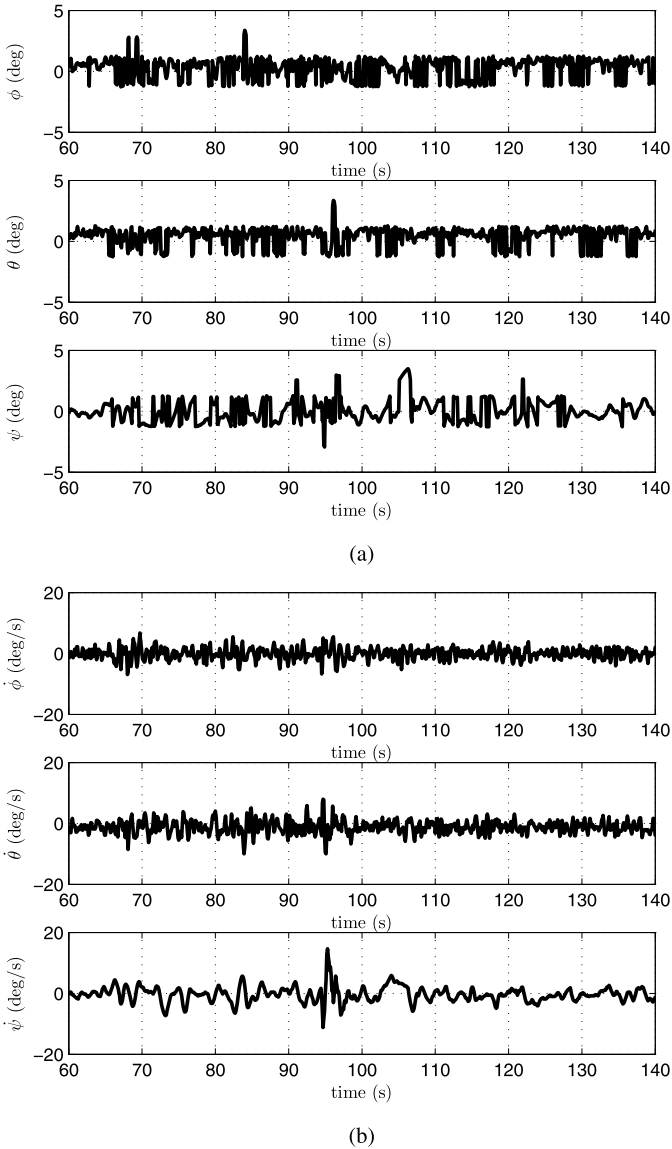
Fig. 3.11 The Cross-Flyer hover flight results: (**a**) Euler angles; (**b**) angular rates

was 15 min, which degrades quickly as new sensors (payload) are added and also if external perturbations are induced. The successful results of this platform motivated the construction of the third vehicle developed during the research activities, which is basically an improved version of the "X-Flyer" quad-rotor.

(a)

(b)
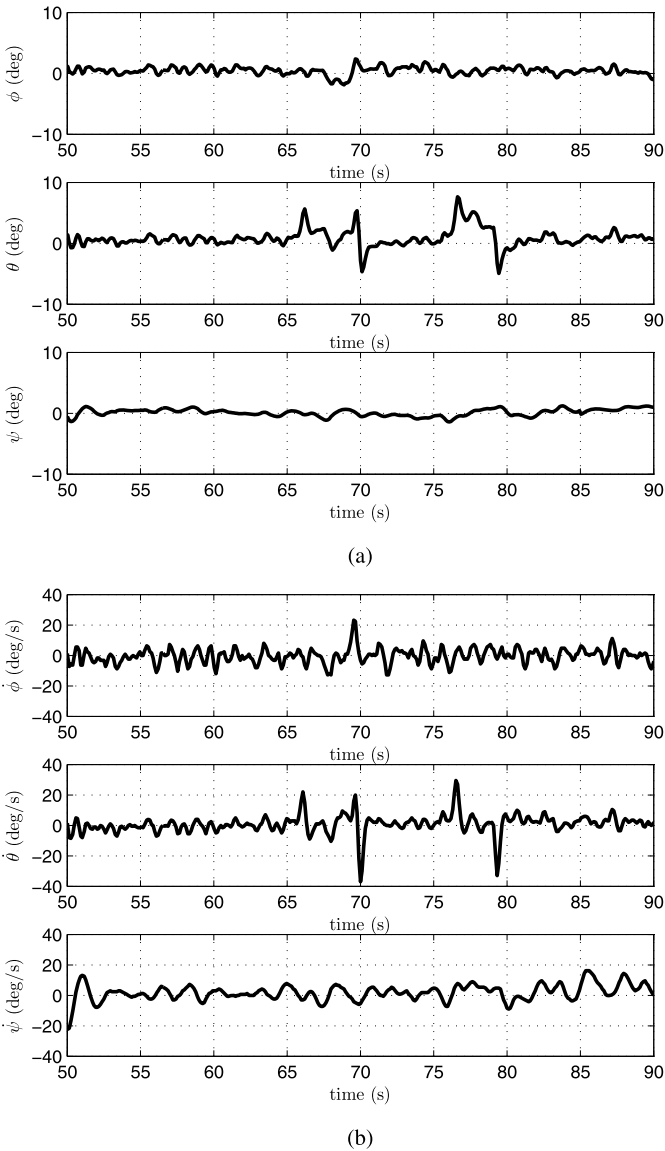
Fig. 3.12 The X-Flyer hover flight results: (**a**) Euler angles; (**b**) angular rates

3.6.3 Improved X-Flyer Hover Graphics

For the construction of the "Improved X-Flyer" platform, the "X-Flayer" motor arrangement was kept. However, as explained in the "Improved X-Flyer" board characteristics, the motor's current sensing stage was omitted, with the purpose of reducing the vehicle's weight and reducing the processing time of the control al-

gorithm. It was decided to install the MIDG II INS/GPS IMU from Microbotics Inc® since it provides better accuracy when measuring angles, in combination with the ADXRS610 MEMS analog angular rate sensors. In addition, this version was equipped with the BL-Outrunner 2827-35 motors from Robbe ROXXY, notably increasing its payload capacity. When using this component combination (motor model, motor arrangement, Microbotics IMU and ADXRS610 gyros), the platform achieves a very responsive attitude stabilization.

The platform tests consisted of autonomous hover flights. External perturbations were also induced in order to verify the vehicle robustness. Figure 3.13(a) shows an induced perturbation in the roll angle of approx. 20° (at 42 seconds). It can be seen that the control algorithm takes the Euler angles near zero very quickly, ensuring a smooth and robust hover flight. The behavior of the angular rates in Fig. 3.13(b) are relatively small and smooth signals. A video of the experiment can be seen in http://www.youtube.com/watch?v=FOXLdQz4IBA.

The maximum time of flight achieved with this vehicle was 15 min. The maximum payload achieved with the "Improved X-Flyer" is 600 grams, which enables carrying more sensors and an extra power supply.

3.7 Concluding Remarks

This chapter described the technological challenges addressed to design an experimental platform consisting of a supervisory ground station and a quad-rotor robotic helicopter with an embedded autopilot. We presented the evolution of the quad-rotor design, driven by specific requirements found during the research activities, but also for the commercial availability of improved technologies.

Experimental results have shown that the three systems developed perform well for hovering flight as well as robustness in presence of external perturbations. In addition, the control strategy developed ensures good stability despite the accuracy of the inertial estimation system. However, when comparing the behavior of the first two developed platforms ("Cross-Flyer and "X-Flyer") with the "Improved X-Flyer", it can be seen that the quad-rotor performance was notably enhanced.

From the experiments, it is considered that keeping Euler angles near zero, as well as having a fast response for correcting external perturbations, are characteristics depending mainly on the next factors:

- Control algorithm working at high rates.
- Sensors with good accuracy.
- Motor characteristics and motors arrangement.

Those characteristics motivated the development of the "Improved X Flayer" platform.

The main interest for developing the quad-rotor vehicle consisted of designing a robotic platform well suited to the specific research needs. The characteristics under consideration were a fast dynamic behavior for correcting Euler angles and a robust

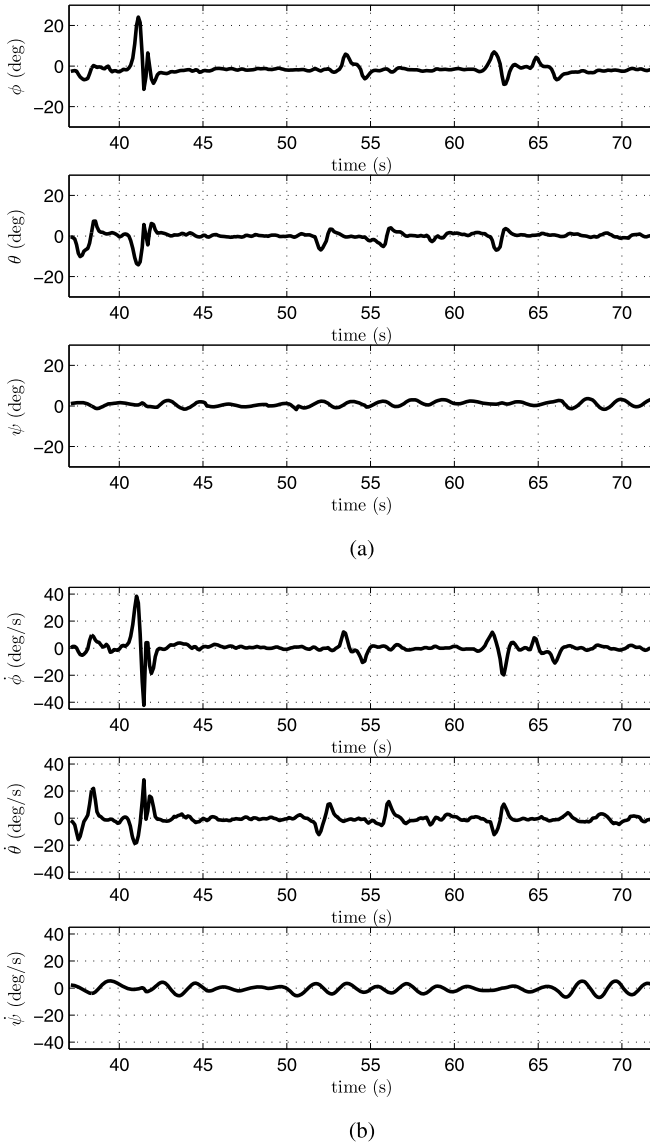**Fig. 3.13** The Improved X-Flyer hover flight results: (**a**) Euler angles; (**b**) angular rates

and reliable communication between the embedded autopilot and the supervisory ground station. Such features facilitate the implementation of additional sensory systems for performing, for example, vision-based positioning and navigation. Subjects related to those tasks will be presented in subsequent chapters.

Chapter 4
Hovering Flight Improvement

This chapter presents the design of an embedded control system for improving the attitude stabilization of a quad-rotor UAV. The control strategy uses low-cost components and includes an extra control loop based on motor armature current feedback. The additional control loop significantly improves the performance of the quad-rotor attitude stability. This technique results in a controller that is robust with respect to external disturbances as has been observed experimentally [71].

4.1 Introduction

UAV applications strongly depend on the fact that the attitude of the aerial vehicle is satisfactorily stabilized. Indeed, if the pitch and roll angles are not controlled with a high degree of precision, the UAV will experience translational drift and it will move away from its desired position.

Few studies have been conducted concerning the improvement of the attitude stabilization for aerial vehicles. Most of the main results have been developed for spacecraft such as satellites [81]. In [56] and [69] the authors proposed a methodology for improving attitude stability using a momentum wheel in a four rotor dual-spin vehicle. Another approach is presented in [49], where the authors propose a sensor fusion technique in order to improve the attitude of an UAV. The approach is based on an extended Kalman filter using information coming from an IMU and a GPS. The approach proposed here studies the attitude stabilization improvement in a different context, it considers a UAV embedded control system that uses not only standard sensors as IMUs and gyros for stabilization, but also the motor armature current. This control strategy improves stabilization of the UAV attitude around the origin. The UAV attitude improvement technique increases the autonomy of the system, opening a new range of applications where a higher performance of the attitude control is required.

It was noticed from standard quad-rotor control systems that the global performance strongly depends on the fact that all the four ESC have the same characteristics. In other words, for a given control input signal, one expects that the four

Fig. 4.1 Electronic
commutator circuit with
BL-DC motor
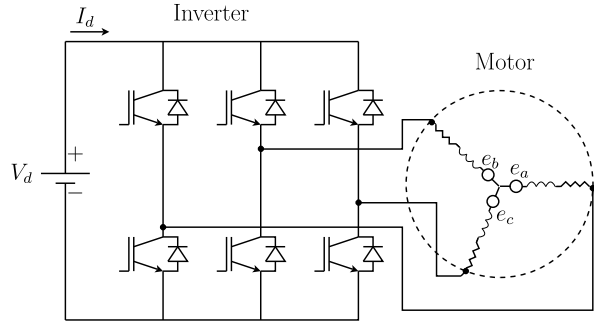
ESCs produce the same rotor speed. This is in general not the case when using off-the-shelf low-cost components. The contribution of the proposed strategy consists of including an extra control loop for each ESC, in such a way that for any control input given, the four ESCs produce almost the same armature current, which will produce the same speed in the four motors. This additional internal control loop significantly improves the performance of the quad-rotor attitude stability.

This chapter is organized as follows. Section 4.2 describes the functioning of brushless direct current (BL-DC) motors and their corresponding speed controllers. The attitude improvement control strategy is given in Sect. 4.3. Section 4.4 presents the embedded control system configuration. Section 4.5 describes the experimental results and finally the concluding remarks are given in Sect. 4.6.

## 4.2 Brushless DC Motor and Electronic Speed Controller

The BL-DC motor, also known as trapezoidal back electromotive force (EMF) motor, is a permanent magnet synchronous machine where the magnetic fields are uniformly distributed in the air gap such that when the motor is turning at constant speed, the back EMF has a trapezoidal shape in time [53, 54]. The BL-DC motor with its permanent magnet field excitation replaces electromagnets, which have windings and require an external electric energy source. Additionally, using an electronic commutator in the form of an inverter, the BL-DC motor replaces the mechanical commutator allowing the armature of the machine to be on the stator, see Figs. 4.1 and 4.2. Those developments enable a better cooling and allow higher voltages to be achieved. The main feature of the BL-DC motor is that, by detecting the rotor position (or magnetic poles), it is possible to generate the inverter control signals for motor speed regulation.

In steady state and neglecting the losses, the power input and the developed torque-speed are related by the following expression:

$$P_e = e_a i_a + e_b i_b + e_c i_c = T_e \omega_m \tag{4.1}$$

where e_* and i_* are the back EFM and the current, respectively, of phases a, b and c, see Fig. 4.1, T_e is the developed torque and ω_m is the rotor's mechanical speed.

Fig. 4.2 Outrunner 1200
BL-DC motor: (**a**) upper
view; (**b**) bottom view

(a)   (b)
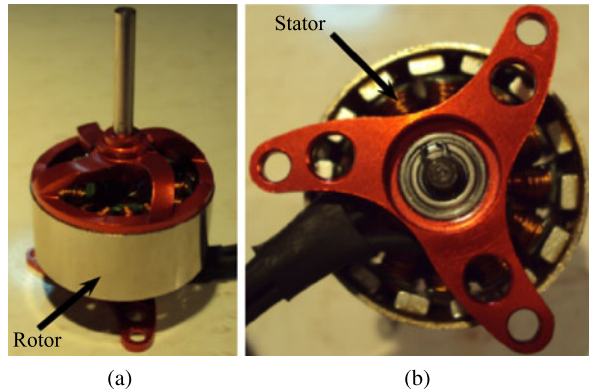
Since only two phases appear in series across the inverter input at any instant, the
power is ideally constant and is given by

$$P_e = 2V_c I_d \qquad (4.2)$$

where V_c is the phase back EFM (for each of the two phases that conduct current)
and I_d is the DC line current. Therefore from (4.1) and (4.2) the developed torque
T_e is given by

$$T_e = \frac{pI_d V_c}{\omega_r} \qquad (4.3)$$

where p is the number of poles and ω_r is the rotor's electrical speed with $\omega_m = 2\omega_r/p$. The back EFM is proportional to the speed as follows:

$$V_c = K_E \omega_r \qquad (4.4)$$

where K_E is the voltage constant depending on the number of turns on the winding
and the magnetic flux, which are constants. Introducing (4.4) in (4.3) one obtains

$$T_e = K_T I_d \qquad (4.5)$$

where K_T is the torque constant with $K_T = pK_E$. Note that in view of (4.1)
and (4.2) the torque and speed of the rotor both depend on the armature current
I_d and the voltage applied to the input terminals. Hence, the armature current gives
information concerning the developed thrust which can be used to improve the atti-
tude performance of the quad-rotor.

A commercial ESC system in a radio controlled (RC) aircraft consists of a three-
phase inverter with rotor-position feedback. The inverter operates as a rotor-position
sensitive electronic commutator, similar to a mechanical commutator in a DC ma-
chine, transforming power from the source to an appropriate form to drive the BL-
DC motor. Based on the rotor position and a speed control signal (in the form of
servo signal) a micro controller computes a control algorithm to determine the gate

Fig. 4.3 RC aircraft ESC:
(**a**) top view; (**b**) bottom view

(a)   (b)
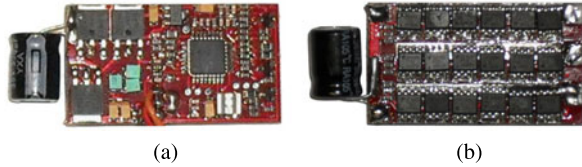
signal for each transistor in the power electronic converter, see Fig. 4.3. The rotor
position of the BL-DC motor is generally determined by using Hall sensors, re-
solvers or absolute position encoders. However, those sensors increase the cost size
and weight of the motor. For this reason, sensorless control of small BL-DC motors
is the most commonly used method in the commercially available ESC. The sensor-
less control technique is based on position sensing by using the motor's back EMF.
In the BL-DC motor, only two of the three phases are excited at any instant, leaving
the third winding floating. The back EMF voltage in the floating winding can be
measured in order to establish a switching strategy to control the rotor speed.

These kinds of ESC are designed in such a way that the desired speed control
signal is measured using an external input. The speed control signal is a PWM sig-
nal (often called servo signal) which is used to drive servo devices. This signal
has a period of 20 ms and varies from 1 ms to 2 ms, depending on the manufac-
turer and type of servo. In a servo device, the angular motion of the shaft can vary
between 0° to 180°. Therefore, the middle position of the shaft occurs when the
pulse width is approximately 1.5 ms, full left rotation of the shaft occurs at 1 ms
pulse width and 2 ms pulse width duration causes the shaft to revolve fully right.
Since the quad-rotor dynamics are relatively fast, the servo signal is not suitable for
practical control purposes. Physically, this means that the motor's response must be
fast enough to generate the forces and torques required for flight maneuvers. Thus,
servo signal alone is not enough for controlling a quad-rotor adequately. Indeed,
servo signal frequency and resolution are not appropriate to achieve a suitable flight
performance. However, using additional information such as the rotor's speed or the
electric power introduced to the motors, it is possible to stabilize the quad-rotor and
improving the vehicle dynamic behavior.

Some experiments were performed on a quad-rotor equipped commercial ESCs.
Based on the results, it has been noticed that for a constant input signal applied si-
multaneously to the four identical ESCs, the motor's speed response were different.
This implies that the forces generated in each axis of the quad-rotor are not symmet-
rical, and therefore there exists a position drift of the quad-rotor in the $x-y$ plane,
see Fig. 2.4. Such a drift is normally compensated by trimming the joystick control
manually. This problem can also be overcome by implementing a control loop using
additional data as mentioned above.

4.3 Control Strategy for Attitude Improvement

A control strategy to improve the attitude stabilization of the quad-rotor is intro-
duced. First, the attitude control strategy is presented, which is based on the angular

rate and position feedback. Next, a control algorithm using the motor's armature current is proposed.

4.3.1 Attitude Control

Let us consider the following state space representation of (2.33)–(2.35) representing the quad-rotor angular dynamics:

$$\dot{\boldsymbol{\Psi}} = A\boldsymbol{\Psi} + \mathbf{B}\tau_\psi$$
$$\dot{\boldsymbol{\Theta}} = A\boldsymbol{\Theta} + \mathbf{B}\tau_\theta \qquad (4.6)$$
$$\dot{\boldsymbol{\Phi}} = A\boldsymbol{\Phi} + \mathbf{B}\tau_\phi$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix}, \qquad \boldsymbol{\Theta} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \qquad \boldsymbol{\Phi} = \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

One obtains the discrete-time system of (4.6) as follows [28]:

$$\boldsymbol{\Psi}(k+1) = \tilde{A}\boldsymbol{\Psi}(k) + \tilde{\mathbf{B}}\tau_\psi(k)$$
$$\boldsymbol{\Theta}(k+1) = \tilde{A}\boldsymbol{\Theta}(k) + \tilde{\mathbf{B}}\tau_\theta(k) \qquad (4.7)$$
$$\boldsymbol{\Phi}(k+1) = \tilde{A}\boldsymbol{\Phi}(k) + \tilde{\mathbf{B}}\tau_\phi(k)$$

where

$$\tilde{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}, \qquad \tilde{\mathbf{B}} = \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix}$$

and T_s is the sampling period. Consider now the following control algorithm [63]:

$$\tau_\psi(k) = -\bar{\mathbf{K}}\bar{\boldsymbol{\Psi}}(k) \qquad (4.8)$$
$$\tau_\theta(k) = -\bar{\mathbf{K}}\bar{\boldsymbol{\Theta}}(k) \qquad (4.9)$$
$$\tau_\phi(k) = -\bar{\mathbf{K}}\bar{\boldsymbol{\Phi}}(k) \qquad (4.10)$$

where

$$\bar{\mathbf{K}} = \begin{bmatrix} \bar{k}_1 & \bar{k}_2 \end{bmatrix}$$

is the state feedback gain,

$$\bar{\boldsymbol{\Psi}}(k) = \begin{bmatrix} \psi_1(k) - \psi_d \\ \psi_2(k) \end{bmatrix}; \qquad \bar{\boldsymbol{\Theta}}(k) = \begin{bmatrix} \theta_1(k) - \theta_d \\ \theta_2(k) \end{bmatrix}; \qquad \bar{\boldsymbol{\Phi}}(k) = \begin{bmatrix} \phi_1(k) - \phi_d \\ \phi_2(k) \end{bmatrix}$$
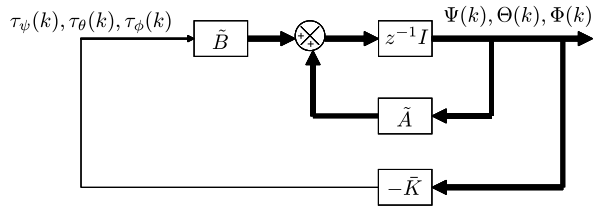
and ψ_d, θ_d, ϕ_d are the desired angular positions coming from the joystick manual
control. Thus

$$\bar{\boldsymbol{\Psi}}(k+1) = (\tilde{A} - \tilde{\mathbf{B}}\bar{\mathbf{K}})\bar{\boldsymbol{\Psi}}(k)$$
$$\bar{\boldsymbol{\Theta}}(k+1) = (\tilde{A} - \tilde{\mathbf{B}}\bar{\mathbf{K}})\bar{\boldsymbol{\Theta}}(k) \qquad\qquad (4.11)$$
$$\bar{\boldsymbol{\Phi}}(k+1) = (\tilde{A} - \tilde{\mathbf{B}}\bar{\mathbf{K}})\bar{\boldsymbol{\Phi}}(k)$$

Choosing $\bar{\mathbf{K}}$ such that the eigenvalues of the matrix $(\tilde{A} - \tilde{\mathbf{B}}\bar{\mathbf{K}})$ lie inside the unit cir-
cle, system (4.11) is asymptotically stable. It is important to point out that gains $\bar{\mathbf{K}}$,
for each state of the system equations in (4.7), were tuned experimentally as in a
classical PID controller. Figure 4.4 shows the block diagram representation of the
attitude control system.

The controllers (4.8), (4.9), and (4.10) are applied to each one of the four ESC
according to the following equations:

$$T_{M_1}(k) = G(k) + \tau_\theta(k) + \tau_\psi(k) \qquad\qquad (4.12)$$

$$T_{M_2}(k) = G(k) - \tau_\theta(k) + \tau_\psi(k) \qquad\qquad (4.13)$$

$$T_{M_3}(k) = G(k) - \tau_\phi(k) - \tau_\psi(k) \qquad\qquad (4.14)$$

$$T_{M_4}(k) = G(k) + \tau_\phi(k) - \tau_\psi(k) \qquad\qquad (4.15)$$

where T_{M_i}, for $i = 1, \ldots, 4$, is the control signal applied to each ESC, and G is the
desired throttle input coming from the joystick, see Fig. 2.1.

Finally, for practical reasons, the desired signals for each motor, represented by
the group of (4.12)–(4.15), are modified to produce the desired speed controls sig-
nals which are sent to the speed controllers. This is done by implementing a linear
interpolation function, resulting in a servo signal that can be driven by the speed
controller, see Fig. 4.5.

4.3.2 Armature Current Control

As was mentioned in Sect. 4.2, each speed controller has its own speed control loop,
but there is no direct connection between the rotor's speed and the main control
of the system (attitude control). In other words, considering the rotor's speed as
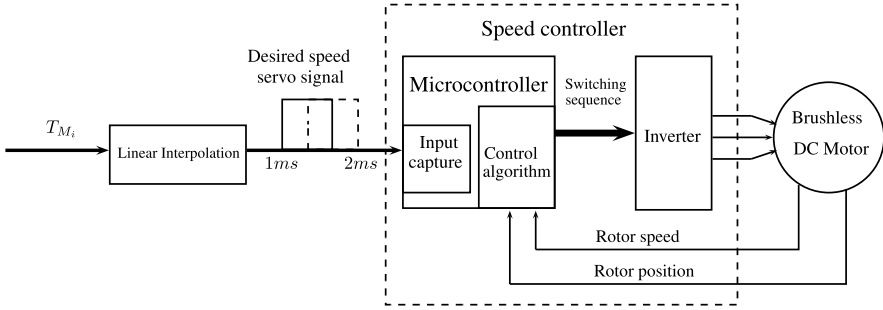
Fig. 4.5 Control input applied to the speed controllers

the output of the system and without measuring it, one has an open-loop control system. For this reason, a control loop using the power measurement for controlling the torque-speed of the rotors is proposed.

For improving the attitude stabilization, an additional controller has been implemented, which uses the DC line current that flows from the battery to the speed controller. This additional information not only improves the attitude stabilization, it also provides robustness with respect to external perturbations. The main idea is to regulate the angular positions of the quad-rotor as well as the DC current of each ESC. This is achieved by comparing the desired signals from (4.12)–(4.15), with respect to the actual value of DC current for each of the four ESCs. A PD controller has been implemented by approximating the relationship between the attitude control signals and the DC currents using a four degree polynomial. The polynomial that fits the current data to an attitude control signal is given by

$$\hat{T}_{M_i}(k) = a_4 I_{d_i}^4(k) + a_3 I_{d_i}^3(k) + a_2 I_{d_i}^2(k) + a_1 I_{d_i}(k) + a_0 \qquad (4.16)$$

where $\hat{T}_{M_i}$ is the conversion to an attitude control signal, I_{d_i} is the DC line current, for $i = 1, \ldots, 4$ (for each of the four speed controllers), and a_0 to a_4 are the coefficients of the polynomial. After performing several experiments, this polynomial fitted well the attitude signal in a least-squares sense. The control algorithm that stabilizes the attitude dynamics including the torque-speed regulation is given by

$$\bar{T}_{M_i}(k) = k_{p_i} e_{M_i}(k) + k_{d_i} \left(\frac{e_{M_i}(k) - e_{M_i}(k-1)}{T_s} \right) \qquad (4.17)$$

where $\bar{T}_{M_i}$ is the feedback control signal applied to each speed controller, also

$$e_{M_i}(k) = T_{M_i}(k) - \hat{T}_{M_i}(k)$$

is the error, and k_{p_i} and k_{d_i} are the proportional and derivative gains, respectively. The resulting control diagram is shown in Fig. 4.6. These additional control parameters were tuned together with the attitude control parameters.
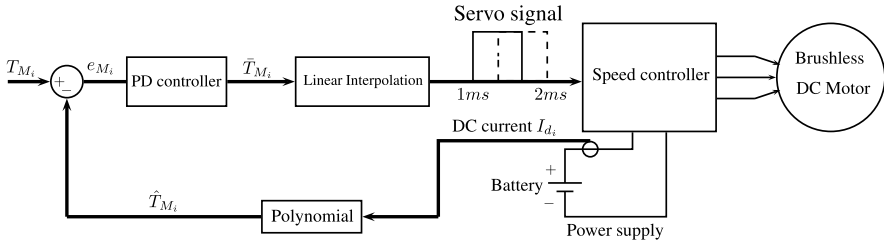
Fig. 4.6 Closed-loop control system using the DC current

4.4 Experimental System Configuration

The experimental platform used to test the improvement of the attitude control is described. The system consists of a quad-rotor, a supervisory ground station, and a wireless data link.

4.4.1 Aerial Vehicle

The quad-rotor used to evaluate the improvement of the attitude control corresponds to the Cross-Flyer experimental platform presented previously in Sect. 3.4.1, Fig. 3.4. As a reminder, the embedded electronic system includes two interconnected boards: the first board is the control unit, while the second is devoted to the motors ESCs and the motor's current sensors. The control unit includes the inertial sensors, and the armature current control runs in the embedded processor to ensure the stabilization of the engine during flight. The second board, shown in Fig. 3.3(b), is the core of the motor current monitoring circuitry, its characteristics can be resumed as follows.

- **Signal conditioning circuitry:** Each motor control input is decoupled from the rest of the electronic system by the signal conditioner. The measured motors current, which are used for feedback control, are also filtered and appropriately conditioned.
- **Motor current monitoring:** This circuit allows the measurement of the current passing through each one of the motors. The current measuring is achieved using a shunt resistor connected to a shunt current monitor. The current monitor output signal is a voltage proportional to the current passing through the motor. This signal is sent to the ADCs of the DSP via the boards connection bus. The objective of this current sensing system is the improvement of the quad-rotor stabilization [71]. As mentioned before, a PD control loop is used to reduce the error between the desired current (attitude control input) and the actual motor's current. This avoids the need for trimming at the beginning of the experiments or when the sensor's temperature changes.

Fig. 4.7 Cross-Flyer UAV and supervisory ground station
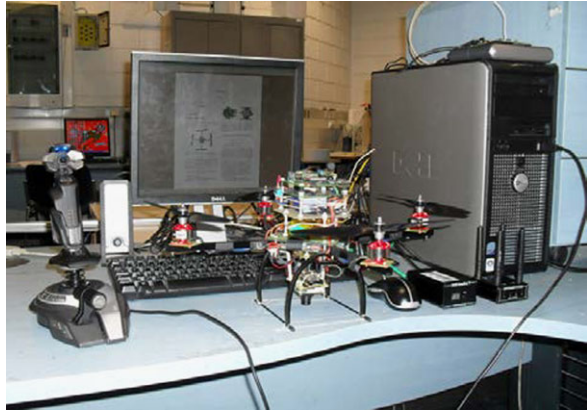
4.4.2 Supervisory Ground Station

The supervisory ground station used during experiments consists of a desktop PC, a flight simulator joystick and a XBee ZB ZigBee PRO radio modem. The ground station sends data generated by the user to the aerial vehicle. In addition, it receives and saves all information needed to debug and analyze the flight experiments and results. Figure 4.7 shows the complete experimental system: UAV and ground station.

4.5 Experimental Results

This section presents real-time experimental results obtained when applying the proposed controller, based on motor armature current feedback, to the Cross-Flyer quad-rotor described in Sect. 4.4. The control parameters adjustment followed standard methods for tuning a PID control loop. Several experiments were conducted with and without the motor armature current feedback (using the same attitude control parameters) for the purpose of comparing the hovering performance. The obtained controller is robust with respect to aggressive external disturbances as has been observed experimentally. Videos from experiments are available online at http://www.youtube.com/watch?v=eyPHxTJrTf0.

As can be seen from Figs. 4.8, 4.9 and 4.10, the proposed controller performs well and improves in practice the attitude stabilization. Notice from Fig. 4.11 that, in hovering, the motor armature current feedback provides better stability even in the presence of aggressive disturbances. Note also that without disturbances, the armature currents of motors pair M_1 and M_3, and pair M_2 and M_4, tends to be the same, as can be seen in Figs. 4.12(a) and 4.12(b), respectively. A picture of the Cross-Flyer flying autonomously during tests is shown in Fig. 4.13.
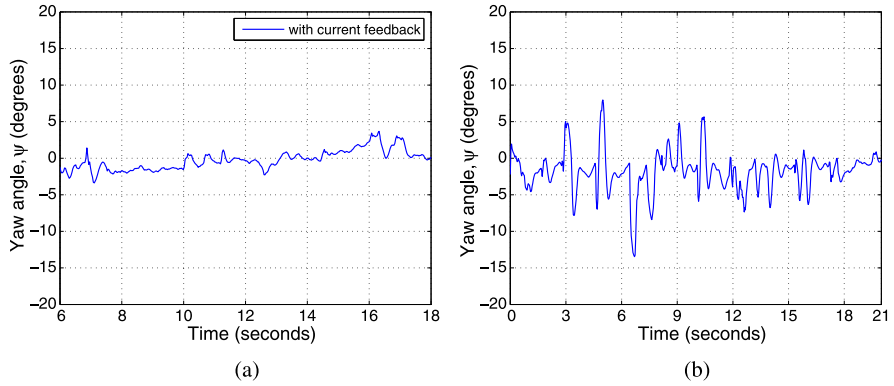
Fig. 4.8 Yaw angle performance: (**a**) with armature current feedback; (**b**) without motor armature current feedback
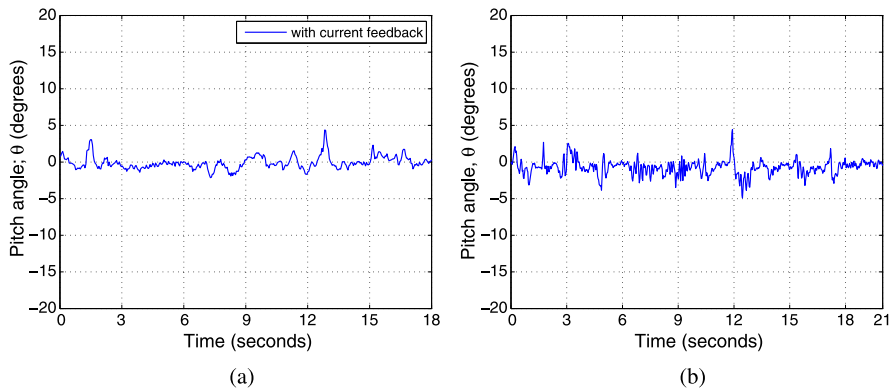
Fig. 4.9 Pitch angle performance: (**a**) with armature current feedback; (**b**) without motor armature current feedback
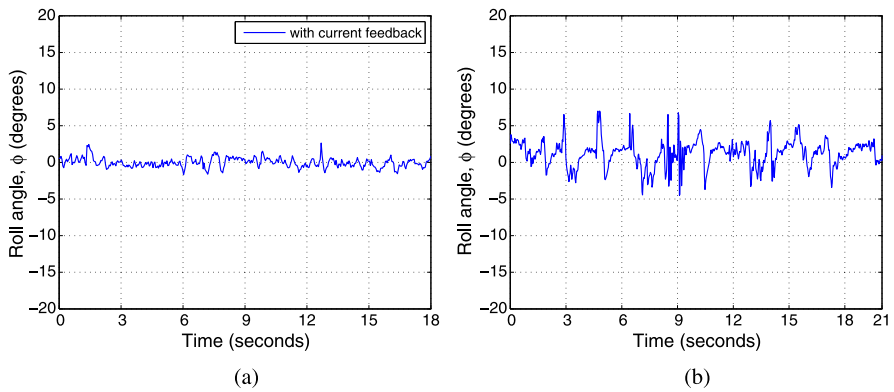
Fig. 4.10 Roll angle performance: (**a**) with armature current feedback; (**b**) without motor armature current feedback

Fig. 4.11 Roll angle performance in the presence of disturbances: (**a**) with armature current feedback; (**b**) without motor armature current feedback
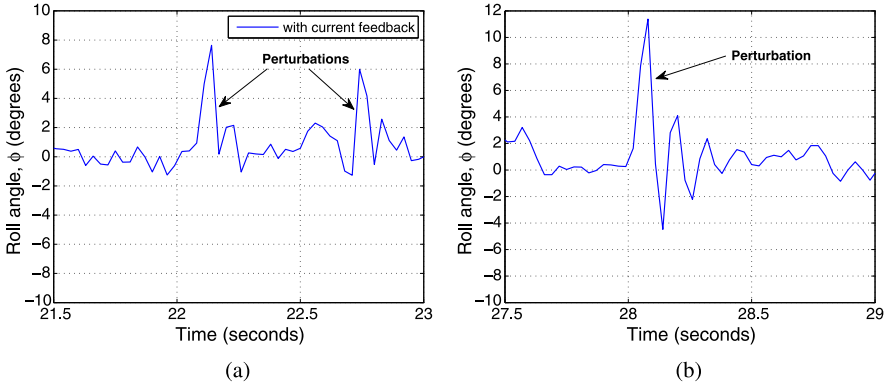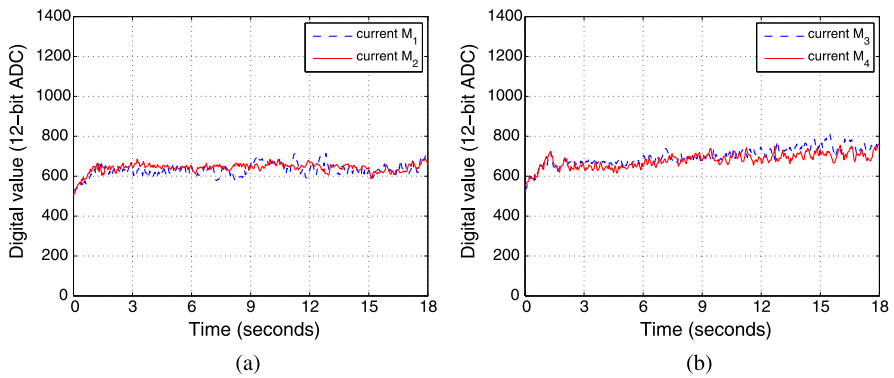

Fig. 4.12 Currents measurements: (**a**) from motors M_1 and M_2; (**b**) from motors M_3 and M_4

Fig. 4.13 The quad-rotor mini-aircraft hovering autonomously
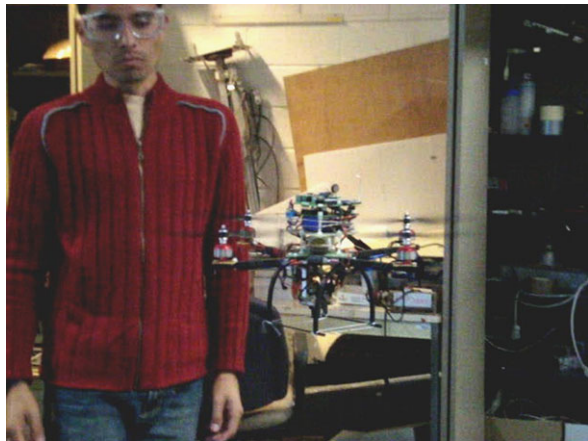
4.6 Concluding Remarks

In this chapter, an attitude control algorithm based on the motor armature current feedback was proposed and applied to a quad-rotor system. The control strategy, based on low-cost components, consists of adding an internal control loop on each ESC in such a way that for any given control input, the four motors turn at almost the same speed. The proposed controller was successfully tested in real-time experiments. The attitude stabilization performance has been considerably improved avoiding drift of the UAV from its desired angular position. In addition, robustness of the proposed controller with respect to external disturbances has been observed experimentally. Given that the quad-rotor Euler angles are very close to the origin, the resulting UAV can be effectively combined with other sensors, like GPS or imaging sensing systems, in order to perform autonomous positioning or trajectory tracking tasks.

Chapter 5
Imaging Sensors for State Estimation

The implementation of imaging systems onboard UAVs allows the development of original methodologies for extracting information concerning the surrounding environment, as well as of the vehicle itself. Imaging sensors, are very attractive since they are passive, non-contact, very versatile, and low-cost. In addition, they can be used in situations where other sensing devices fail, leading to a whole new group of applications that can be accomplished. However, before using imaging sensors a mathematical model must be computed, describing how the 3-dimensional points are represented in 2-dimensional images. For the interested reader, there are numerous computer vision books covering related topics in great detail, see for example [40] and [60]. In the present chapter the required theoretical background for the construction of imaging models is briefly presented, and in addition, the physical implementation of the imaging system for estimating the sates of a UAV is also addressed.

This chapter is divided as follows. Section 5.1 presents the pinhole camera model, as well as a camera calibration procedure for obtaining the intrinsic parameters. Next, stereo imaging is introduced in Sect. 5.2, with an explanation of the epipolar geometry concept. Also, a method for stereo calibration and rectification is presented. With the purpose of allowing the estimation of relative translational speed using an imaging sensor, the concept of optical flow and a method for its computation are detailed in Sect. 5.3. In Sect. 5.4, some important points that must be considered when implementing an imaging system onboard a quad-rotor UAV are discussed. In addition, the development of both a monocular and a stereo imaging system is presented, as well as the software architecture conceived with the purpose of estimating the data required for performing vision-based tasks. Finally, some concluding remarks are presented in Sect. 5.5.

5.1 Camera Model

Consider the central projection of 3-dimensional points in space onto a plane. Let the center of projection be the center of an Euclidean coordinate system, and consider

Fig. 5.1 Pinhole camera geometry. C is the camera center, p is the principal point. The camera center is placed at the coordinate origin. Note the image plane is placed in front of the camera center
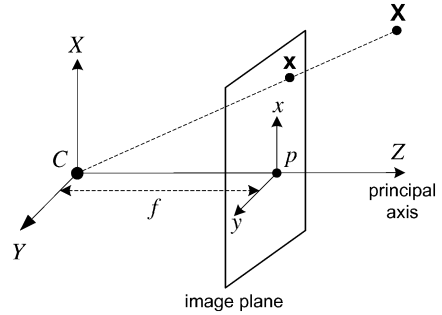
the plane $Z = f$, which is known as the *image plane* or *focal plane*. Under the pinhole camera model, a point in space with coordinates $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ is mapped to the point in the image plane where a line joining the point $\mathbf{X}$ to the center of projection meets the image plane, see Fig. 5.1. By similar triangles, it could be verified that the point $(X, Y, Z)^{\mathrm{T}}$ is mapped to the point $(f\frac{X}{Z}, f\frac{Y}{Z}, f)^{\mathrm{T}}$ in the image plane. Ignoring the image last coordinate, the central projection mapping from world to image coordinates is described as

$$(X, Y, Z)^{\mathrm{T}} \mapsto \left(f\frac{X}{Z}, f\frac{Y}{Z} \right)^{\mathrm{T}} \tag{5.1}$$

This is defined as a mapping from an 3-dimensional Euclidean space $\mathbb{R}^3$ to a 2-dimensional Euclidean space $\mathbb{R}^2$. The center of projection C is called the *camera center*, also known as the *optical center*. The line from the camera center perpendicular to the image plane is called the *principal axis* or *principal ray* of the camera, and the point where the principal axis meets the image plane is called the *principal point* and is represented by p. The plane through the camera center parallel to the image plane is called the *principal plane* of the camera [40].

Central Projection Using Homogeneous Coordinates   If the world and image points are represented by homogeneous vectors, then central projection is simply expressed as a linear mapping between their homogeneous coordinates. In particular, equation (5.1) may be written in terms of matrix multiplication as

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{5.2}$$

The 3×4 matrix in (5.2) may be written as $\mathrm{diag}(f, f, 1)[I|\mathbf{0}]$ where $\mathrm{diag}(f, f, 1)$ is a diagonal matrix and $[I|\mathbf{0}]$ represents a matrix divided up into a 3×3 block (the identity matrix) plus a column vector, here the zero vector. Let $\mathbf{X}$ be the notation for the world points represented by the homogeneous 4-dimensional vector $(X, Y, Z, 1)^{\mathrm{T}}$. Also, let $\mathbf{x}$ be the image points represented by a homogeneous 3-dimensional vector, and P for the 3×4 homogeneous *camera projection matrix*.

Fig. 5.2 Image (x, y) and camera (x_{cam}, y_{cam}) coordinates systems
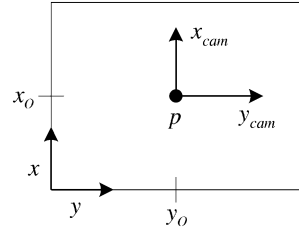
Then, (5.2) is written compactly as

$$\mathbf{x} = P\mathbf{X} \tag{5.3}$$

Equation (5.3) defines the camera matrix for the pinhole model of central projection [40]

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.4}$$

Principal Point Offset   Equation (5.1) assumes that the coordinates origin in the image plane is placed at the principal point. In practice, it may not be, so that in general there is a mapping

$$(X, Y, Z)^{\mathrm{T}} \mapsto \left(f\frac{X}{Z} + p_x, f\frac{Y}{Z} + p_y \right)^{\mathrm{T}} \tag{5.5}$$

where $(p_x, p_y)^{\mathrm{T}}$ are the coordinates of the principal point p, see Fig. 5.2. Expressing (5.5) in homogeneous coordinates one has

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{5.6}$$

Writing

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{5.7}$$

equation (5.6) has the form

$$\mathbf{x} = K[I|\mathbf{0}]\mathbf{q} \tag{5.8}$$

Matrix K is called the *camera calibration matrix*. In (5.8), $(X, Y, Z, 1)^{\mathrm{T}}$ has been written as $\mathbf{q}$ to emphasize that the camera is assumed to be located at the origin of a Euclidean coordinate system, with the principal axis of the camera pointing straight down the Z-axis, and the point $\mathbf{q}$ is expressed in this coordinate system. Such a coordinate system is called the *camera coordinate frame*.

Extrinsic Parameters   In general, points in space will be expressed in terms of a different Euclidean coordinate frame, known as the *world coordinate frame*. The world and camera coordinate frames are related via a rotation and a translation. If $\tilde{\mathbf{X}} \in \mathbb{R}^{3 \times 1}$ is an inhomogeneous vector representing the coordinates of a point in the world coordinate frame, and $\tilde{\mathbf{q}}$ represents the same point in the camera coordinate frame, then we may write $\tilde{\mathbf{q}} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$, where $\tilde{\mathbf{C}}$ represents the coordinates of the camera center in the world coordinate frame, and $R \in \mathbb{R}^{3 \times 3}$ is a rotation matrix representing the orientation of the camera coordinates frame. This equation can be written in homogeneous coordinates as

$$\mathbf{q} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X} \tag{5.9}$$

Putting together (5.8) and (5.9) leads to

$$\mathbf{x} = K R [I \, | - \tilde{\mathbf{C}}] \mathbf{X} \tag{5.10}$$

where $\mathbf{X}$ is now in a world coordinate frame. This is the general mapping given by a pinhole camera. One sees that a general pinhole camera $P = K R [I - \tilde{\mathbf{C}}]$, has nine degrees of freedom: three for K (the elements f, p_x, p_y), three for R, and three for $\tilde{\mathbf{C}}$. The parameters contained in K are called the *internal camera parameters*, or the *internal orientation* of the camera. The parameters of R and $\tilde{\mathbf{C}}$ which relate the camera orientation and position to a world coordinate system are called the *extrinsic parameters* or the *exterior orientation*. It is often convenient not to make the camera center explicit, and instead to represent the world to image transformation as $\tilde{\mathbf{q}} = R\tilde{\mathbf{X}} + \mathbf{t}$. In this case the camera matrix is simply

$$P = K[R|\mathbf{t}] \tag{5.11}$$

where from (5.10)

$$\mathbf{t} = -R\tilde{\mathbf{C}} \tag{5.12}$$

Intrinsic Properties   The pinhole camera model just derived assumes that the image coordinates are Euclidean coordinates, having equal scales in both axial directions. In the case of charge-coupled device (CCD) cameras, there is the possibility of having non-square pixels. If image coordinates are measured in pixels, this has the extra effect of introducing unequal scale factors in each direction. In particular, if the number of pixels per unit distance in image coordinates are m_x and m_y in the x and y directions, respectively, then the transformation from world coordinates to pixel coordinates is obtained by multiplying (5.7) on the left by an extra factor $\text{diag}(m_x, m_y, 1)$. Thus, the general form of the calibration matrix of a CCD camera is

$$K = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{5.13}$$

where $\alpha_x = f m_x$ and $\alpha_y = f m_y$ represent the focal length of the camera in terms of pixel dimensions in the x and y direction, respectively. Similarly, $\tilde{\mathbf{x}}_0 = (x_0, y_0)$ is the principal point in terms of pixel dimensions, with coordinates $x_0 = m_x p_x$ and $y_0 = m_y p_y$. A CCD camera thus has 10 degrees of freedom.

Consider now a camera calibration matrix K of the form

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \tag{5.14}$$

The added parameter s is referred to as the *skew parameter*, which will be zero for most normal cameras [40]. A camera

$$P = K R [I | - \tilde{\mathbf{C}}] \tag{5.15}$$

for which the calibration matrix is of the form in (5.14) is called a *finite projective camera*. A finite projective camera has 11 degrees of freedom. This is the same number of degrees of freedom as a 3×4 matrix, defined up to an arbitrary scale. The left hand 3×3 sub-matrix of P, equal to KR, is non-singular. Conversely, any 3×4 matrix P for which the left hand 3×3 sub-matrix is non-singular is the camera matrix of some finite projective camera, because P can be decomposed as $P = K R [I | - \tilde{\mathbf{C}}]$. Indeed, letting M be the left 3×3 sub-matrix of P, one decomposes M as a product $M = K R$ where K is upper-triangular of the form of (5.14), and R is a rotation matrix. This decomposition is essentially the RQ matrix decomposition. The camera matrix P can therefore be written $P = M[I | M^{-1} \mathbf{p}_4] = K R [I | - \tilde{\mathbf{C}}]$ where $\mathbf{p}_4$ is the last column of P.

The previous relationships fully describe how a general projective camera P maps 3-dimensional world points $\mathbf{X}$, to 2-dimensional image points $\mathbf{x}$, according to $\mathbf{x} = P\mathbf{X}$. However, cameras use lenses, which distort the images, more or less, depending on their design. In particular, lenses with a wide field of view tend to significantly distort images. Therefore, camera model has to be extended with a distortion model to compensate distortion before performing any image processing. For the distortion model, radial and tangential distortion must me assumed. Let $\mathbf{x}$ denote the projected coordinates of a point $\mathbf{X}$ before the multiplication with the camera calibration matrix K, and let $\mathbf{x}_d$ represent its corresponding distorted coordinates. The distortion model can be written as

$$\mathbf{x}_d = \underbrace{\mathbf{x}(1 + k_1 r^2 + k_2 r^4)}_{\text{radial}} + \underbrace{\begin{bmatrix} 2k_3 xy + k_4(r^2 + 2x^2) \\ k_3(r^2 + 2y^2) + 2k_4 xy \end{bmatrix}}_{\text{tangential}} \tag{5.16}$$

with $r^2 = x^2 + y^2$ and the distortion factor $L(d) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4$, considering the k_* factors as part of the interior calibration of the camera. The distortion model implies that nonlinear equations have to be solved in order to recover $\mathbf{x}$. Due to this, when image processing involves real-time applications, it is useful to pre-compute a look-up table that maps distorted to undistorted coordinates.

5.1.1 Camera Calibration

This section describes a method for estimating the camera projection matrix from corresponding 3-dimensional world and 2-dimensional image entities. The simplest such correspondence is that between a 3-dimensional point $\mathbf{X}$ and its corresponding 2-dimensional image $\mathbf{x}$ under the unknown camera mapping P. Given sufficiently many correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$, the camera matrix P may be determined. The internal parameters K of the camera may be extracted from the matrix P by means of an RQ decomposition.

Consider a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3-dimensional points $\mathbf{X}_i$ and 2-dimensional image points $\mathbf{x}_i$. A camera matrix P must be found, namely a 3×4 matrix such that $\mathbf{x}_i = P\mathbf{X}_i$ for all i. $\mathbf{x}_i = P\mathbf{X}_i$ involves homogeneous coordinates thus $\mathbf{x}_i$ and $P\mathbf{X}_i$ just have to be proportional (defined up to a scalar factor). Therefore, it is possible to use the cross product $\mathbf{x}_i \times P\mathbf{X}_i = 0$. Let $\mathbf{p}_1^{\mathsf{T}}$, $\mathbf{p}_2^{\mathsf{T}}$, $\mathbf{p}_3^{\mathsf{T}}$ be the three row vectors of P

$$
P\mathbf{X}_i = \begin{bmatrix} \mathbf{p}_1^{\mathsf{T}}\mathbf{X}_i \\ \mathbf{p}_2^{\mathsf{T}}\mathbf{X}_i \\ \mathbf{p}_3^{\mathsf{T}}\mathbf{X}_i \end{bmatrix}; \quad \mathbf{x}_i \times P\mathbf{X}_i = \begin{bmatrix} y_i\mathbf{p}_3^{\mathsf{T}}\mathbf{X}_i - w_i\mathbf{p}_2^{\mathsf{T}}\mathbf{X}_i \\ w_i\mathbf{p}_1^{\mathsf{T}}\mathbf{X}_i - x_i\mathbf{p}_3^{\mathsf{T}}\mathbf{X}_i \\ x_i\mathbf{p}_2^{\mathsf{T}}\mathbf{X}_i - y_i\mathbf{p}_1^{\mathsf{T}}\mathbf{X}_i \end{bmatrix}
$$

for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ there exists a relationship

$$
\begin{bmatrix} \mathbf{0}^{\mathsf{T}} & -w_i\mathbf{X}_i^{\mathsf{T}} & y_i\mathbf{X}_i^{\mathsf{T}} \\ w_i\mathbf{X}_i^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & -x_i\mathbf{X}_i^{\mathsf{T}} \\ -y_i\mathbf{X}_i^{\mathsf{T}} & x_i\mathbf{X}_i^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = 0 \tag{5.17}
$$

where each $\mathbf{p}_i^{\mathsf{T}}$ is a 4-vector, the ith row of P. Since the three equations of (5.17) are linearly dependent, one may choose to use only the first two equations

$$
\begin{bmatrix} \mathbf{0}^{\mathsf{T}} & -w_i\mathbf{X}_i^{\mathsf{T}} & y_i\mathbf{X}_i^{\mathsf{T}} \\ w_i\mathbf{X}_i^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}} & -x_i\mathbf{X}_i^{\mathsf{T}} \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = 0 \tag{5.18}
$$

One obtains two independent equations in 11 unknowns (ignoring scale). From a set of n point correspondences (6 as minimum), we obtain a $2n \times 12$ matrix A, by stacking up (5.18) for each correspondence. The projection matrix P is computed by solving the set of equations $A\mathbf{p} = 0$, where $\mathbf{p}$ is the vector containing the entries of the matrix P.

Calibration Using a Chessboard   A common calibration method consists of placing in front of the camera an object with certain number of points whose coordinates are well known in a 3-dimensional reference frame. In principle, any appropriately characterized object could be used as a calibration object, yet the practical choice is a regular pattern such as a chessboard. Some calibration methods in the literature rely on 3-dimensional objects (e.g., a box covered with markers), but flat chessboard patterns are much easier to deal with than 3-dimensional calibration
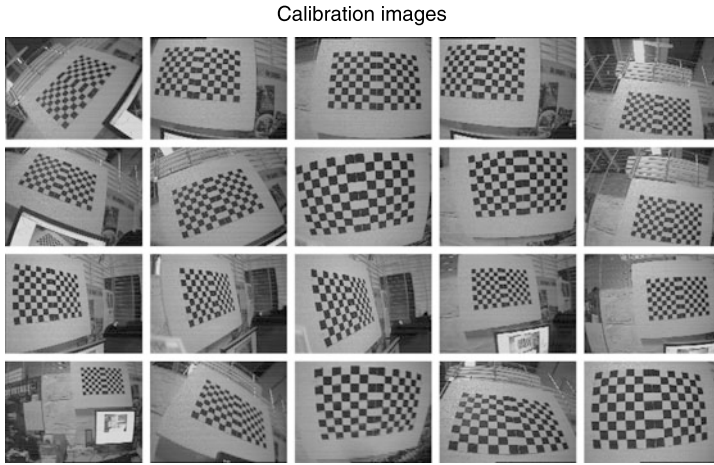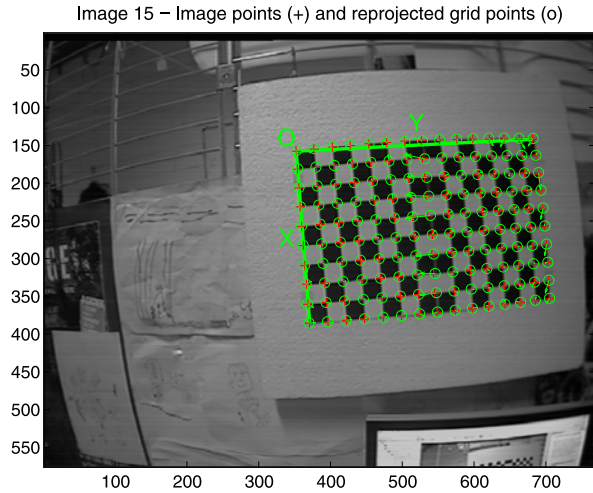
Calibration images

Fig. 5.3 The *Camera Calibration Toolbox for Matlab* GUI displaying the chessboard images

objects. For the previous reason, a chessboard pattern has been built, consisting of alternating black and white squares of 30 mm each.

A calibration technique that requires the camera to observe a planar chessboard shown at a few (at least two) different positions and orientations is presented in [83]. The algorithm takes advantage of the chessboard pattern to compute the square's corners points, then, it extracts the projective transformation between the image points of the n different images, up to a scale factor. With this method, the intrinsic and extrinsic parameters of the camera are computed, in addition, radial lens distortion is modeled as well. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion.

A handy implementation of the previously presented calibration method is the *Camera Calibration Toolbox for Matlab* [17]. This program, available online for downloading and installing, has shown to be a very powerful and easy to use tool, capable of accurately compute the intrinsic and extrinsic parameters of a camera. For the present research, the *Camera Calibration Toolbox for Matlab* was used to calibrate a CTDM-5351 high definition camera from Sony, with a resolution of 752×582 pixels. To obtain a good accuracy, 20 pictures of the chessboard pattern have been taken in different positions and orientations. Figure 5.3 shows the toolbox graphical user interface (GUI) displaying the set of pictures. To perform an accurate calibration, the toolbox requires the number of squares along the horizontal and vertical directions, as well as the individual dimension of each square. In addition, for each chessboard image, one has to manually detect the four external corners of the chessboard. After repeating this process for all the n images, calibration is performed. As a practical example, the internal camera parameters matrix K for the CTDM-5351 high definition camera, as computed by the toolbox, is presented in (5.19):

Fig. 5.4 Chessboard image
photo with highlighted
corners

Optionally, one can reproject on the chessboard images the corners used for cali-
bration. Figure 5.4 shows a chessboard image photo with highlighted corners. From
calibration, extrinsic parameters (rotation and translation) for each chessboard view
are also computed. Figure 5.5(a) shows extrinsic parameters with coordinates cen-
tered in the camera, while Fig. 5.5(b) represent them in a world reference frame.

The camera calibration toolbox provides also a distortion model to compensate
distortion caused by camera lenses. Figure 5.6 shows the resulting distortion model,
considering radial and tangential distortion.

To demonstrate how distortion affects the camera's image, Fig. 5.7(a) shows an
image as provided directly from camera, where straight lines appear curved. After
applying the obtained model to correct radian and tangential distortion, the resulting
image is shown in Fig. 5.7(b).

5.2 Stereo Imaging

Computers accomplish stereo imaging by finding correspondences between points
that are seen by one imager and the same points as seen by the other imager. Then,
the 3-dimensional location of the points can be estimated using such correspon-
dences and a known baseline separation (geometry) between cameras, In practice,
stereo imaging involves four steps:

- *Undistortion*: mathematically remove radial and tangential lens distortion. Ob-
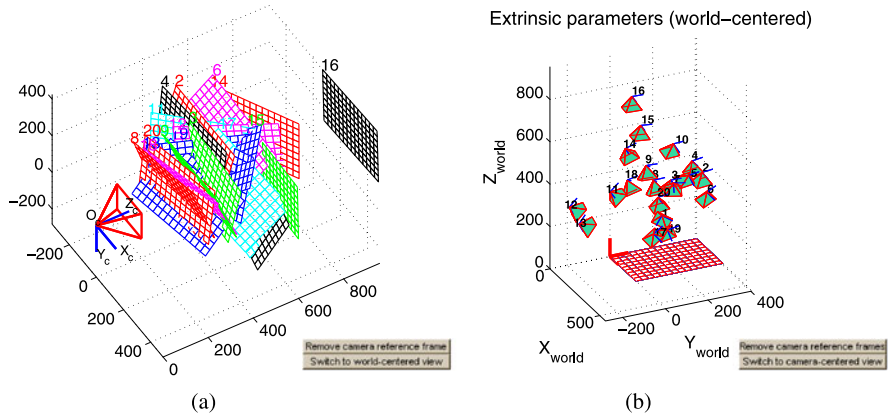 tention of undistorted images.

Fig. 5.4 Chessboard image
photo with highlighted
corners

Image 15 − Image points (+) and reprojected grid points (o)

$$K = \begin{bmatrix} 615.33303 & 0.0 & 377.68498 \\ & 614.07367 & 289.88436 \\ & & 1 \end{bmatrix} \quad (5.19)$$

Optionally, one can reproject on the chessboard images the corners used for cali-
bration. Figure 5.4 shows a chessboard image photo with highlighted corners. From
calibration, extrinsic parameters (rotation and translation) for each chessboard view
are also computed. Figure 5.5(a) shows extrinsic parameters with coordinates cen-
tered in the camera, while Fig. 5.5(b) represent them in a world reference frame.

The camera calibration toolbox provides also a distortion model to compensate
distortion caused by camera lenses. Figure 5.6 shows the resulting distortion model,
considering radial and tangential distortion.

To demonstrate how distortion affects the camera's image, Fig. 5.7(a) shows an
image as provided directly from camera, where straight lines appear curved. After
applying the obtained model to correct radian and tangential distortion, the resulting
image is shown in Fig. 5.7(b).

5.2 Stereo Imaging

Computers accomplish stereo imaging by finding correspondences between points
that are seen by one imager and the same points as seen by the other imager. Then,
the 3-dimensional location of the points can be estimated using such correspon-
dences and a known baseline separation (geometry) between cameras, In practice,
stereo imaging involves four steps:

- *Undistortion*: mathematically remove radial and tangential lens distortion. Ob-
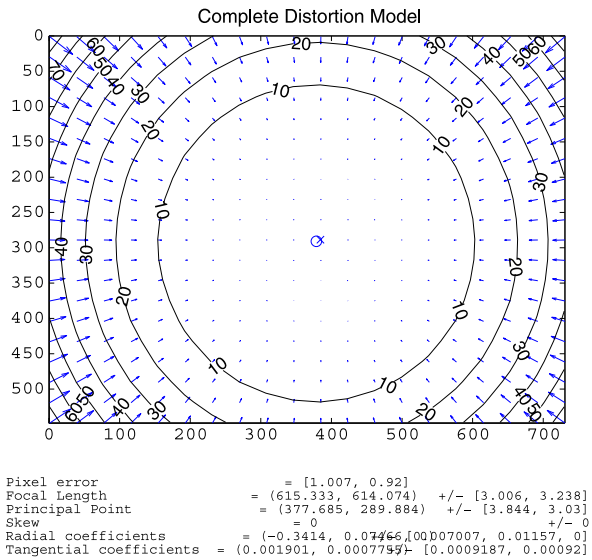 tention of undistorted images.

Fig. 5.5 Extrinsic parameters: (**a**) coordinates centered in the camera; (**b**) world reference frame

Fig. 5.6 Distortion model for the CTDM-5351 camera

Pixel error	= [1.007, 0.92]
Focal Length	= (615.333, 614.074) +/− [3.006, 3.238]
Principal Point	= (377.685, 289.884) +/− [3.844, 3.03]
Skew	= 0 +/− 0
Radial coefficients	= (−0.3414, 0.07466, 0) +/− [0.007007, 0.01157, 0]
Tangential coefficients	= (0.001901, 0.0007755) +/− [0.0009187, 0.00092]

- *Rectification*: adjust for the angles and distances between cameras. Obtention of images that are row-aligned and rectified.
- *Correspondence*: finding the same features in the left and right camera views. Obtention of a disparity map, where the disparities are the differences in x-coordinates on the image planes of the **X** feature viewed in the left and right cameras: $\mathbf{x}_l - \mathbf{x}_r$.
- *Reprojection*: Knowing the geometric arrangement of the cameras, the disparity map can be turned into distances using *triangulation*. Obtention of a depth map.
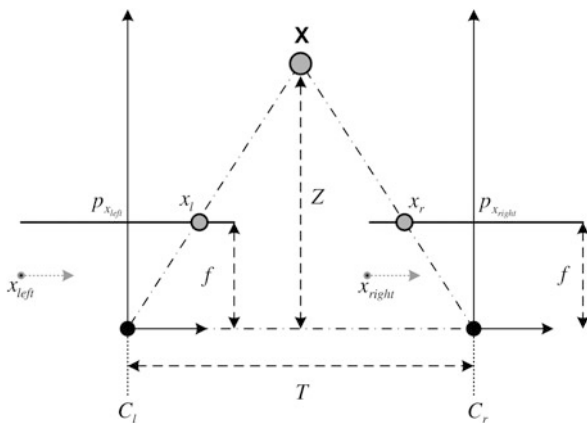
(a)   (b)
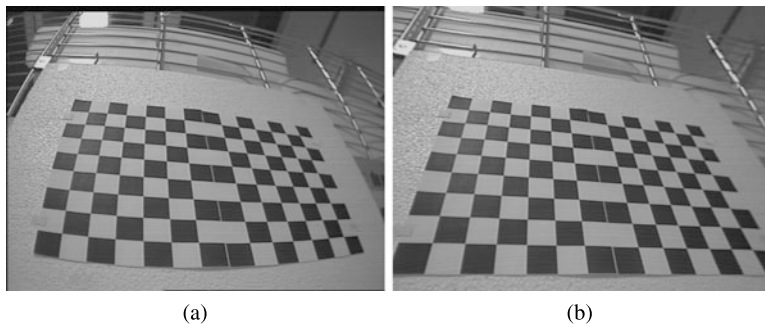
Fig. 5.7 Correcting distortion: (**a**) image as provided directly from camera; (**b**) corrected image

Fig. 5.8 Ideal stereo rig: The depth Z can be found by similar triangles. The principal rays of the imagers begin at the centers of projection C_l and C_r and extend through the principal points of the two image planes at $p_{x_{\text{left}}}$ and $p_{x_{\text{right}}}$


Triangulation   Assume a perfectly undistorted, aligned, and measured stereo rig as shown in Fig. 5.8. Let us further assume a frontal parallel camera arrangement: the images are row-aligned and every pixel row of one camera aligns exactly with the corresponding row in the other camera. Consider one can find a point $\mathbf{X}$ in the physical world in the left and the right image views at $\mathbf{x}_l$ and $\mathbf{x}_r$, which will have the respective horizontal coordinates x_l and x_r. Taking x_l and x_r to be the horizontal positions of $\mathbf{x}_l$ and $\mathbf{x}_r$, respectively, allows to show that the depth is inversely proportional to the disparity between these views. The disparity is defined by $d = x_l - x_r$. From the schema shown in Fig. 5.8, one can easily derive the depth Z by using similar triangles

$$\frac{\mathbf{T} - (x_l - x_r)}{Z - f} = \frac{\mathbf{T}}{Z} \quad \Rightarrow \quad Z = \frac{f\mathbf{T}}{x_l - x_r} \tag{5.20}$$

There is obviously a nonlinear relationship between depth and disparity. When disparity is near 0, small disparity differences make for large depth differences. When disparity is large, small disparity differences do not change the depth by much. As a consequence, stereo vision systems have high depth resolution only for objects relatively near the camera.

Fig. 5.9 Stereo coordinate system: the pixel coordinates are relative to the upper left corner of the image, and the two planes are row-aligned; the camera coordinates are relative to the left camera's center of projection
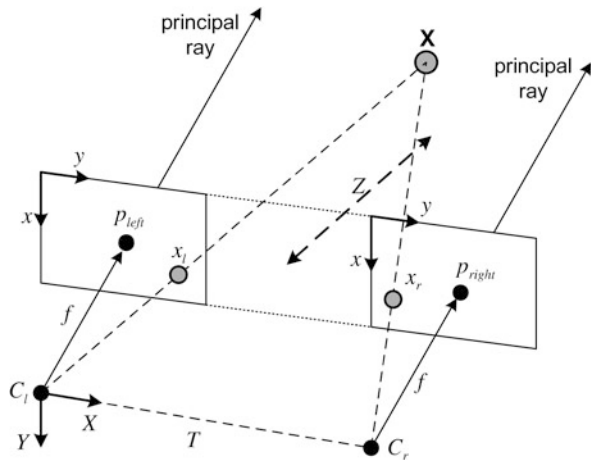
With the purpose of adapting the notation of this work to the notation used by the *Camera Calibration Toolbox for Matlab* [17] let us use a right-handed coordinate system as shown in Fig. 5.9. The left and right imager pixels have image origins at upper left in the image. Pixels are denoted by coordinates (x_l, y_l) and (x_r, y_r), respectively. The center of projection are at C_l and C_r; with principal rays intersecting the image plane at the principal point (p_x, p_y). After rectification, the cameras are row-aligned, displaced from one another by a distance **T**, and with the same focal length f. With this arrangement it is relatively easily to solve for distance. In the real world, cameras will almost never be exactly aligned in the frontal parallel configuration, one must mathematically find image projections and distortion maps that will rectify the left and right images into such arrangement. When building a stereo rig, the cameras must be installed approximately frontal parallel and horizontally aligned, in order to make the mathematical transformations more tractable. If the cameras are not approximately aligned, the mathematical alignment can produce extreme image distortions, reducing the stereo overlap area of the resulting images.

The pair of cameras must capture their images at the exact same time, to avoid having problems if anything is moving in the scene, including the cameras themselves, therefore, the stereo cameras must be synchronized. Failing to do so, one limits oneself to using stereo vision with stationary cameras, viewing static scenes. Figure 5.10 shows the real situation between two cameras and the desired mathematical alignment. In order to perform this mathematical alignment, let us introduce some notations concerning the geometry of two cameras viewing a scene.
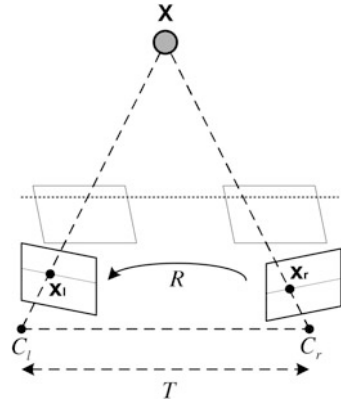
5.2.1 Epipolar Geometry

The basic geometry of a stereo imaging system is shown in Fig. 5.11. It is commonly referred to as *epipolar geometry*. For each camera there is a center of projection, C_l
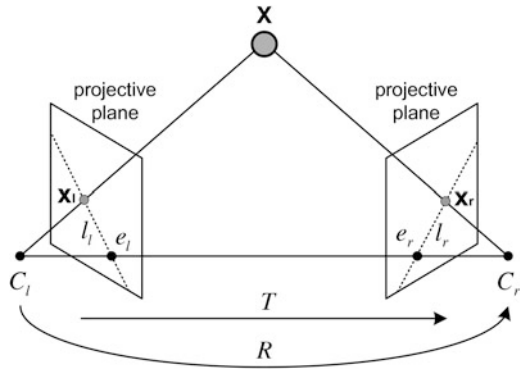
and C_r, and a pair of corresponding projective planes, P_l and P_r. The point $\mathbf{X}$ in the physical world has a projection onto each of the projective planes at $\mathbf{x}_l$ and $\mathbf{x}_r$, respectively. The new points of interest are the epipoles. An epipole e_l (respectively, e_r) on image plane P_l (respectively, P_r) is defined as the image of the center of projection of the other camera C_r (respectively, C_l). The plane in space formed by the viewed point $\mathbf{X}$ and the two epipoles e_l and e_r (or, equivalently, through the two centers of projection C_r and C_l) is called the epipolar plane, and the lines $\mathbf{x}_l e_l$ and $\mathbf{x}_r e_r$ (from the points of projection to the corresponding epipolar points) are called the epipolar lines.

Let us take for example the point $\mathbf{X}$ as seen by the camera on the right. Since that camera sees only $\mathbf{x}_r$ (the projection of $\mathbf{X}$ onto P_r), the actual point $\mathbf{x}_r$ could be located anywhere on the line defined by $\mathbf{x}_r$ and C_r. Such line, obviously, contains $\mathbf{X}$, as well as other points. Furthermore, its projection onto the left image plane P_l is the epipolar line defined by $\mathbf{x}_l$ and e_l. One concludes that the image of all of the possible locations of a point seen in one imager is the line that goes through the corresponding point and the epipolar point on the other imager. Let us summarize some facts about stereo camera epipolar geometry:

- Every 3-dimensional point in view is contained in an epipolar plane that intersects each image in an epipolar line.
- Given a feature in one image, its matching view in the other image must lie along the corresponding epipolar line (epipolar constraint).
- Once the epipolar geometry of the stereo rig is known, the 2-dimensional search for matching features across two imagers becomes a 1-dimensional search along the epipolar lines. This generates computational savings, and allows to reject a lot of points that could otherwise lead to spurious correspondences.
- Order is preserved. If points A and B are visible in both images and occur horizontally in that order in one imager, then they occur horizontally in that order in the other imager.

The Essential and Fundamental Matrices The essential matrix E contains information about the translation and rotation that relate the two cameras in physical space, see Fig. 5.11. E is purely geometrical and knows nothing about imagers. It relates the location, in physical coordinates, of the point $\mathbf{X}$ as seen by the left camera to the location of the same point as seen by the right camera (for which the notation $\mathbf{q}_l$ and $\mathbf{q}_r$ will be used).

The fundamental matrix F contains the same information as E, in addition to information about the intrinsics of both cameras. Therefore, F relates the points on the image plane of one camera in image coordinates (pixels) to the points on the image plane of the other camera in image coordinates (i.e., it relates $\mathbf{x}_l$ to $\mathbf{x}_r$).

Essential Matrix Math Given a point $\mathbf{X}$, one searches to derive a relationship between the observed locations $\mathbf{x}_l$ and $\mathbf{q}_r$ of $\mathbf{X}$ on the two imagers. This relationship is the essential matrix E. Let us use coordinates centered on C_l. In these coordinates, the location of the observed point is $\mathbf{X}_l$ and the origin of the other camera is located at $\mathbf{T}$. The point $\mathbf{X}$ as seen by the right camera is $\mathbf{X}_r$ in that camera's coordinates, where $\mathbf{X}_r = R[\mathbf{X}_l - \mathbf{T}]$. The key step is the introduction of the epipolar plane, which relates all of these things. Let us recall that the equation for all points $\mathbf{x}$ on a plane with normal vector $\mathbf{n}$ and passing through point $\mathbf{a}$ obeys the constraint $(\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} = 0$.

The epipolar plane contains the vectors $\mathbf{X}_l$ and $\mathbf{T}$, thus, a vector $\mathbf{X}_l \times \mathbf{T}$ perpendicular to both can be used instead of $\mathbf{n}$ in the plane equation. Therefore, en equation for all possible points $\mathbf{X}_l$ through the point $\mathbf{T}$ and containing both vectors would be $[\mathbf{X}_l - \mathbf{T}]^{\mathrm{T}}[\mathbf{T} \times \mathbf{X}_l] = 0$. The objective is to relate $\mathbf{x}_l$ and $\mathbf{x}_r$ by first relating $\mathbf{X}_l$ and $\mathbf{X}_r$. Let us draw $\mathbf{X}_r$ into the picture via the equality $\mathbf{X}_r = R[\mathbf{X}_l - \mathbf{T}]$, which can be rewritten as $[\mathbf{X}_l - \mathbf{T}] = R^{-1}\mathbf{X}_r$. Making this substitution and given that $R^{\mathrm{T}} = R^{-1}$ yields $[R^{\mathrm{T}}\mathbf{X}_r]^{\mathrm{T}}[\mathbf{T} \times \mathbf{X}_l] = 0$. Since it is possible to rewrite a cross product as a matrix multiplication, let the matrix S be such that

$$\mathbf{T} \times \mathbf{X}_l = S\mathbf{X}_l = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \mathbf{X}_l \qquad (5.21)$$

Making this substitution for the cross product gives $\mathbf{X}_r^{\mathrm{T}} R S \mathbf{X}_l = 0$. The essential matrix E is defined by the product RS, leading to the compact equation $\mathbf{X}_r^{\mathrm{T}} E \mathbf{X}_l = 0$. In

order to obtain the relation between the points as one observes them on the images, let us substitute using the projection equations

$$\mathbf{q}_l = f_l \frac{\mathbf{X}_l}{Z_l}; \qquad \mathbf{q}_r = f_r \frac{\mathbf{X}_r}{Z_r}$$

and then divide them by $\frac{Z_l Z_r}{f_l f_r}$ to obtain the final result:

$$\mathbf{q}_r^\mathrm{T} E \mathbf{q}_l = 0 \tag{5.22}$$

The 3×3 essential matrix E has rank 2 (rank-deficient matrix), which actually ends up being an equation for a line. There are five parameters in E: three for rotation and two for the direction of translation (scale is not set), along with two additional constraints:

- The determinant is 0 because it is rank-deficient.
- Its two nonzero singular values are equal because the matrix S is skew-symmetric and R is a rotation matrix.

Yielding a total of seven constraints.

Fundamental Matrix Math   In order to find a relationship between a pixel in one image and the corresponding epipolar line in the other image, one has to introduce intrinsic information about the two cameras. To do this, for $\mathbf{x}$ (the pixel coordinate) one substitutes $\mathbf{q}$ and the camera intrinsics matrix that relates them. Recall that $\mathbf{x} = K\mathbf{q}$ (where K is the camera intrinsics matrix) or, equivalently, $\mathbf{q} = K^{-1}\mathbf{x}$. Hence the equation for E becomes

$$\mathbf{x}_r^\mathrm{T} K_r^{-\mathrm{T}} E K_l^{-1} \mathbf{x}_l = 0$$

by defining the fundamental matrix F as $K_r^{-\mathrm{T}} E K_l^{-1}$ now one can write

$$\mathbf{x}_r^\mathrm{T} F \mathbf{x}_l = 0 \tag{5.23}$$

Whereas E operates in physical coordinates, the fundamental matrix F operates in image pixel coordinates. F is of rank 2 and has seven parameters, two for each epipole and three for the homography that relates the two image planes (the scale aspect is missing from the usual four parameters).

Epipolar Lines   Equation (5.23) is true, because if points $\mathbf{x}_l$ and $\mathbf{x}_r$ correspond, then $\mathbf{x}_r$ lies on the epipolar line $l_r = F\mathbf{x}_l$ corresponding to the point $\mathbf{x}_l$. In other words $0 = \mathbf{x}_r^\mathrm{T} l_r = \mathbf{x}_r^\mathrm{T} F \mathbf{x}_l$. Conversely, if image points satisfy the relation $\mathbf{x}_r^\mathrm{T} F \mathbf{x}_l = 0$ then the rays defined by these points are coplanar. This is a necessary condition for points to correspond.

5.2.2 Calibration of the Stereo Imaging System

Stereo calibration is the process of computing the geometrical relationship between the two cameras in space. It depends on finding the rotation matrix R and translation

vector $\mathbf{T}$ that relate the right camera to the left camera, as depicted in Fig. 5.11. For any given 3-dimensional point $\mathbf{X}$, one can separately use a single-camera calibration to put $\mathbf{X}$ in the camera coordinates $\mathbf{X}_l = R_l\mathbf{X} + \mathbf{T}_l$ and $\mathbf{X}_r = R_r\mathbf{X} + \mathbf{T}_r$ for the left and right cameras, respectively. Note in Fig. 5.11 that both views of $\mathbf{X}$ are related by $\mathbf{X}_l = R^{\mathrm{T}}(\mathbf{X}_r - \mathbf{T})$, where R and $\mathbf{T}$ are, respectively, the rotation matrix and translation vector between the cameras. Taking these three equations and solving for the rotation and translation separately yields the following simple relations:

$$R = R_r R_l^{\mathrm{T}} \tag{5.24}$$

$$\mathbf{T} = \mathbf{T}_r - R\mathbf{T}_l \tag{5.25}$$

Given a set of paired views of chessboard corners, the *Camera Calibration Toolbox for Matlab* solves for rotation and translation parameters of the chessboard views, for each camera separately. It then plugs these left and right rotation and translation solutions into (5.24) and (5.25) to solve for the rotation and translation parameters between the two cameras. Image noise and rounding errors cause that each chessboard pair results in slightly different values for R and $\mathbf{T}$. The calibration routine takes the median values for R and $\mathbf{T}$ as an initial approximation of the true solution, then, it runs a robust Levenberg–Marquardt iterative algorithm to find the minimum of the reprojection error of the chessboard corners for both camera views, and the solution for R and $\mathbf{T}$ is returned.

Stereo calibration gives the rotation matrix that will put the right camera in the same plane as the camera, this makes the two image planes coplanar but not row-aligned; to do so, a stereo rectification must be accomplished.

Stereo Rectification   It is easiest to compute the stereo disparity when the two image planes align exactly. Unfortunately, when using a real stereo system, a perfectly aligned configuration is rare since the two cameras almost never have exactly coplanar, row-aligned imaging planes. Figure 5.10 shows the goal of stereo rectification. One seeks to reproject the image planes of the two cameras so that they reside in the exact same plane, with image rows perfectly aligned into a frontal parallel configuration.

One wants the image rows between the two cameras to be aligned after rectification, so that stereo correspondence will be more reliable and computationally tractable. By having to search only one row for a match with a point in the other image, reliability and computational efficiency are both enhanced. The result of aligning horizontal rows within a common image plane containing each image is that the epipoles themselves are then located at infinity. That is, the image of the center of projection in one image is parallel to the other image plane. Since there are an infinite number of possible frontal parallel planes to choose from, one needs to add more constraints, like maximizing view overlap and minimizing distortion.

Eight terms will result from the process of aligning the two image planes. For each camera one obtains a distortion vector, a rotation matrix, and the rectified and unrectified camera matrices (K_{rect} and K, respectively). From these terms, a map for interpolating pixels from the original image can be constructed, in order to create a new rectified image. To compute the rectification terms, the *Camera Calibration*

Toolbox for Matlab implements Bouguet's algorithm, which uses the rotation and translation parameters from two calibrated cameras viewing a calibration pattern.

Calibrated Stereo Rectification: Bouguet's Algorithm   Given R and $\mathbf{T}$ between the stereo images, Bouguet's algorithm attempts to minimize the amount of change reprojection produces for each of the two images while maximizing common viewing area. To minimize image reprojection distortion, the rotation matrix R that rotates the right camera's image plane into the left camera's image plane is split in half between the two cameras, resulting r_l and r_r rotation matrices for the left and right camera, respectively.

Each camera rotates half a rotation, so their principal rays each end up parallel to the vector sum of where their original principal rays had been pointing, putting the cameras into coplanar alignment but not into row alignment. To compute the R_{rect} that will take the left camera's epipole to infinity and align the epipolar lines horizontally, one creates a rotation matrix by starting with the direction of the epipole $\mathbf{e}_l$ itself. Taking the principal point (p_x, p_y) as the left image's origin, the direction of the epipole is directly along the translation vector between the two camera's centers of projection. The next vector $\mathbf{e}_2$, must be orthogonal to $\mathbf{e}_1$ but is otherwise unconstrained. For $\mathbf{e}_2$, choosing a direction orthogonal to the principal ray is a good choice. This is accomplished by using the cross product of $\mathbf{e}_1$ with the direction of the principal ray and then normalizing, so that obtaining another unit vector. The $\mathbf{e}_3$ is just orthogonal to $\mathbf{e}_1$ and $\mathbf{e}_2$, it can be found using their cross product. Then, $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ are expressed as

$$\mathbf{e}_1 = \frac{T}{\|T\|}; \qquad \mathbf{e}_2 = \frac{[-T_y \ T_x \ 0]^{\mathrm{T}}}{\sqrt{T_x^2 + T_y^2}}; \qquad \mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$$

The matrix that takes the epipole in the left camera to infinity is then

$$R_{\text{rect}} = \begin{bmatrix} \mathbf{e}_1^{\mathrm{T}} \\ \mathbf{e}_2^{\mathrm{T}} \\ \mathbf{e}_3^{\mathrm{T}} \end{bmatrix}$$

which rotates the left camera about the center of projection so that the epipolar lines become horizontal and the epipoles are at infinity. The row alignment of the two cameras is achieved by setting $R_l = R_{\text{rect}} r_l$ and $R_r = R_{\text{rect}} r_r$. The rectified left and right camera matrices $K_{l_{\text{rect}}}$ and $K_{r_{\text{rect}}}$ are also computed, but returned combined with projection matrices Π_l and Π_r:

$$\Pi_l = K_{l_{\text{rect}}} \Pi_l' = \begin{bmatrix} \alpha_{x_l} & s_l & x_{0_l} \\ 0 & \alpha_{y_l} & y_{0_l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.26}$$

$$\Pi_r = K_{r_{\text{rect}}} \Pi_r' = \begin{bmatrix} \alpha_{x_r} & s_r & x_{0_r} \\ 0 & \alpha_{y_r} & y_{0_r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.27}$$
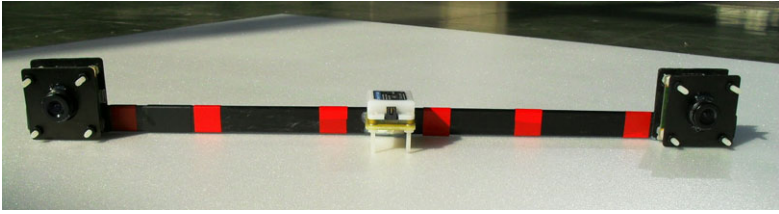
Fig. 5.12 The stereo rig: uEye UI-1226LE-M-G cameras installed with a separation of 35 cm

The projection matrices take a 3-dimensional point in homogeneous coordinates to a 2-dimensional point in homogeneous coordinates as

$$
\Pi \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \tag{5.28}
$$

Screen coordinates can be calculated as $(x/w, y/w)$. 2-dimensional points can also then be reprojected into three dimensions given their screen coordinates and the camera intrinsics matrix. The reprojection matrix is defined as

$$
Q = \begin{bmatrix} 1 & 0 & 0 & -x_{0_l} \\ 0 & 1 & 0 & -y_{0_l} \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & \frac{(x_{0_l}-x_{0_r})}{T_x} \end{bmatrix} \tag{5.29}
$$

If the principal rays intersect at infinity, then $x_{0_l} = x_{0_r}$ inf and the term in the lower right corner is 0. Given a 2-dimensional homogeneous point and its associated disparity d, the point can be projected into three dimensions by

$$
Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{5.30}
$$

The 3-dimensional coordinates are then $(X/W, Y/W, Z/W)$. Bouguet's rectification method yields the ideal stereo configuration shown in Fig. 5.8. New image centers and new image bounds are then chosen for the rotated images so as to maximize the overlapping viewing area.

Calibrating a Stereo Rig Using the *Camera Calibration Toolbox for Matlab*
A stereo rig has been constructed with the purpose of performing stereo imaging studies. Two *uEye UI-1226LE-M-G* cameras from *IDS GmbH* [48] have been installed with a separation of 35 cm between them, as shown in Fig. 5.12. Bouguet's stereo calibration method *Camera Calibration Toolbox for Matlab* requires first that both cameras are individually calibrated. For this purpose, the method previously presented in Sect. 5.1.1 has been used to obtain the calibration parameters of the
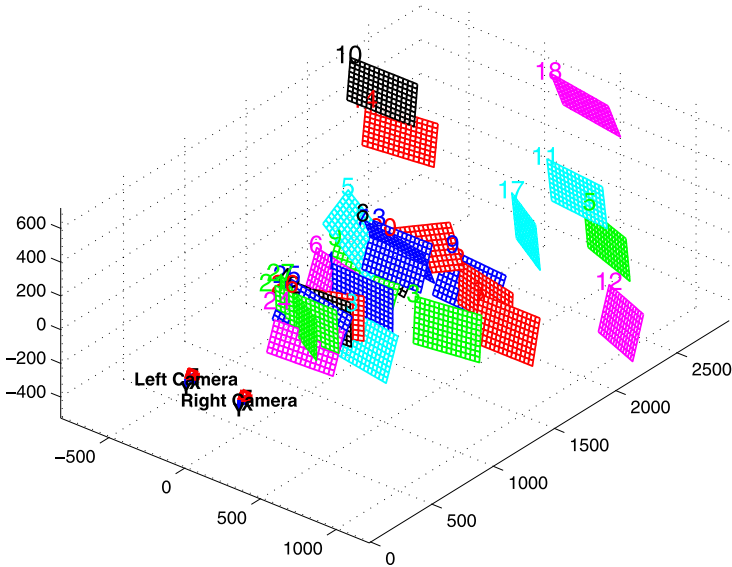
Fig. 5.13 The stereo rig and the chessboard positions

pair of uEye cameras. Using those parameters, the toolbox estimates the position of
the chessboard pattern with respect to the stereo rig, and generates an image to il-
lustrate them, it can be seen in Fig. 5.13. Once calibration is performed, the toolbox
provides the next data,

$$K_l = \begin{bmatrix} 369.30485 & 0.0 & 173.42147 \\ 0.0 & 372.40078 & 139.26367 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{5.31}$$

$$K_r = \begin{bmatrix} 368.41268 & 0.0 & 200.22282 \\ 0.0 & 369.24829 & 138.03250 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{5.32}$$

$$R = \begin{bmatrix} 0.9969 & 0.0057 & -0.0778 \\ -0.0072 & 0.9998 & -0.0185 \\ 0.07775 & 0.0191 & 0.9967 \end{bmatrix} \tag{5.33}$$

$$\mathbf{T} = \begin{bmatrix} -354.9561 & 9.3886 & -19.6539 \end{bmatrix}^{\mathrm{T}} \tag{5.34}$$

where K_l and K_r represent the left and right camera matrices, respectively, R is
the rotation matrix that rotates the right camera's image plane into the left camera's
image plane, and $\mathbf{T}$ is the translation vector that relates the right camera with the
left camera. The essential matrix is then described by

$$E = \begin{bmatrix} 0.6175 & 47.2370 & 11.4110 \\ -19.4754 & 6.4582 & 354.8174 \\ -9.7339 & -352.2883 & 7.5109 \end{bmatrix} \tag{5.35}$$

(a)   (b)

(c)   (d)
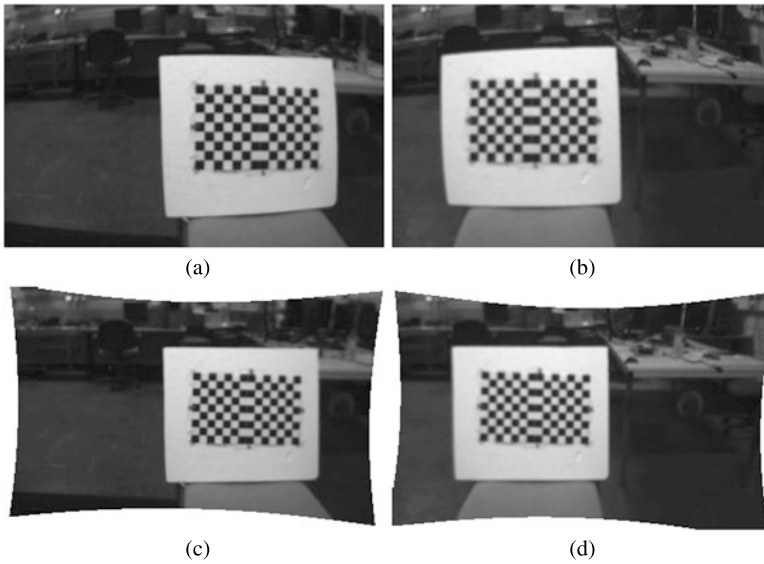
Fig. 5.14 Stereo pairs: (**a**) left image; (**b**) right image; (**c**) rectified left image; (**d**) rectified right image

and the fundamental matrix is

$$F = \begin{bmatrix} 0.000004 & 0.000344 & -0.017762 \\ -0.000142 & 0.000046 & 0.979145 \\ -0.007552 & -1.021412 & 12.22679 \end{bmatrix} \tag{5.36}$$

The obtained projection matrices for the left and right cameras are then

$$\Pi_l = \begin{bmatrix} 369.30485 & 0.0 & 173.42147 & 0.0 \\ 0.0 & 372.40078 & 139.26367 & 0.0 \\ 0.0 & 0.0 & 1.00000 & 0.0 \end{bmatrix} \tag{5.37}$$

$$\Pi_r = \begin{bmatrix} 382.855 & 5.9537 & 170.8889 & -134705.5132 \\ 8.0565 & 371.8116 & 130.7228 & 753.8591 \\ 0.0777 & 0.0191 & 0.9967 & -19.6539 \end{bmatrix} \tag{5.38}$$

Figure 5.14(a)–(b) shows one of the left and right images pairs used during the stereo calibration process. Applying the calibration parameters obtained allows rectifying the image pair, the result is shown in Fig. 5.14(c)–(d). It can be verified that after rectification, the overlapping viewing area has been maximized. Some 3-dimensional reconstructions have been performed with the purpose of validating the accuracy of the stereo calibration. The four corners of the white square surrounding the chessboard have been selected as the desired features to reconstruct. From the upper left corner, and in a clock-wise rotation, the four corners are located in the left and right images, respectively, at
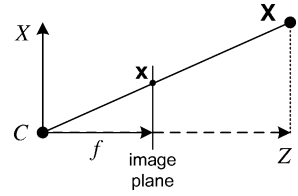
$$\mathbf{C}_{1_{\text{left}}} = \begin{bmatrix} 153 \\ 49 \end{bmatrix}; \qquad \mathbf{C}_{1_{\text{right}}} = \begin{bmatrix} 49 \\ 51 \end{bmatrix}$$

$$\mathbf{C}_{2_{\text{left}}} = \begin{bmatrix} 324 \\ 53 \end{bmatrix}; \qquad \mathbf{C}_{2_{\text{right}}} = \begin{bmatrix} 221 \\ 41 \end{bmatrix}$$

$$\mathbf{C}_{3_{\text{left}}} = \begin{bmatrix} 327 \\ 197 \end{bmatrix}; \qquad \mathbf{C}_{3_{\text{right}}} = \begin{bmatrix} 224 \\ 194 \end{bmatrix}$$

$$\mathbf{C}_{4_{\text{left}}} = \begin{bmatrix} 159 \\ 208 \end{bmatrix}; \qquad \mathbf{C}_{4_{\text{right}}} = \begin{bmatrix} 54 \\ 197 \end{bmatrix}$$

The 3-dimensional reconstruction of each corner, respectively, result is

$$\mathbf{C}_1 = \begin{bmatrix} -0.2496 \\ -0.3113 \\ 1.3552 \end{bmatrix}$$

$$\mathbf{C}_2 = \begin{bmatrix} 0.3586 \\ -0.3146 \\ 1.3093 \end{bmatrix}$$

$$\mathbf{C}_3 = \begin{bmatrix} 0.3650 \\ 0.2083 \\ 1.2980 \end{bmatrix}$$

$$\mathbf{C}_3 = \begin{bmatrix} -0.2267 \\ 0.2305 \\ 1.3163 \end{bmatrix}$$

with values expressed in meters. The 3-dimensional reconstruction is performed
with satisfactory results, since the real position of the chessboard during the exper-
iment was 1.3 m in front of the cameras. Note also how the (x, y) estimations of
each corner are consistent with their corresponding 3-dimensional position.

Using the mathematical relationships just derived, the stereo vision system can
be installed onboard an autonomous agent with the purpose of estimating its 3-
dimensional position with respect to its surrounding environment.

5.3 Optical Flow

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a
visual scene caused by the relative motion between an observer (the imaging sen-

sor), and the scene. The optical flow is created by the translational and the rotational movements of a point $\mathbf{X} = (X, Y, Z)$ on the camera reference frame. Consider the projection of $\mathbf{X}$ into the image plane, $\mathbf{x} = (x, y)$, as shown in Fig. 5.15, its time derivative in function of the point $\mathbf{X}$ can be expressed as

$$\frac{d\mathbf{x}}{dt} = \frac{1}{Z} \frac{d\mathbf{X}}{dt} \tag{5.39}$$

with

$$\frac{d\mathbf{X}}{dt} = \mathbf{V} + R \times \mathbf{X} \tag{5.40}$$

where $\mathbf{V}$ represents the translational movement and R the rotational movement of the point. Then, optical flow can be expressed as

$$\begin{bmatrix} \text{OF}_x \\ \text{OF}_y \end{bmatrix} = \mathbf{T}_{\text{OF}} + \mathbf{R}_{\text{OF}} \tag{5.41}$$

with the translational part

$$\mathbf{T}_{\text{OF}} = \frac{1}{Z} \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{5.42}$$

and the rotational part

$$\mathbf{R}_{\text{OF}} = \begin{bmatrix} \frac{xy}{f} & -(f + \frac{x^2}{f}) & y \\ (f + \frac{y^2}{f}) & -\frac{xy}{f} & -x \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{5.43}$$

where OF_* is the optical flow component in the (x or y) coordinate of the point $\mathbf{x}$, V_* and ω_* are the translational and rotation velocities rates, respectively, of the point $\mathbf{X}$. Equation (5.41) represents the optical flow defined on the image plane. However, optical flow can also be defined over other projection surfaces, such as a sphere, which is often used for its passivity properties. If we apply the appropriate mathematical relationships when using a different projection surface, optical flow information is accurately estimated.

5.3.1 Computing Methods

The standard techniques for calculating the optical flow can be classified in four main groups: differential or gradient methods based on intensity [45, 58], methods of correlation or block matching [7, 75], methods based on energy [42] and those based on phase [36, 80]. Block matching techniques present good accuracy and performance against aperture problems and large displacements, unfortunately they are computationally expensive, their accuracy decreases in the presence of deformation, and displacements of less than one pixel are not detectable. Energy-based and phase-based methods are computationally expensive and their implementation

is very complicated. Due to the derivatives estimation, differential methods present high sensitivity issues caused by poor lighting or image noise. However, they have good precision and their processing is less time consuming than all the other cited techniques. From all the discussed methods, the differential approach is the most implemented. Indeed, differential methods are well known and widely used in the literature for the calculation of optical flow.

The optical flow differential method is based on the computation of the spatial-temporal derivatives of image intensity over a large image region (global method) or local neighborhoods (local methods). Let $I(x, t)$ be a 1-dimensional image, and suppose the intensity remains the same, except if it shifts right or left at constant velocity V_x. Therefore, the spatial derivative $I_x = \frac{\partial I}{\partial x}$ and the temporal derivative $I_t = \frac{\partial I}{\partial t}$ follow the rule

$$I_t = -V_x I_x \tag{5.44}$$

For 2-dimensional images, assuming that the intensity $I(x, y, t)$ structure is constant in a local time-varying region, one writes

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{5.45}$$

where t is the time and $(\delta x, \delta y)$ is the image displacement after time δt. From (5.45) it is possible to obtain the constraint of conservation of intensity, expressed by

$$\nabla I(x, y)(v_x, v_y) + I_t = 0 \tag{5.46}$$

In [58] the authors have constructed a technique for estimating the optical flow, which is known as the Lucas–Kanade algorithm. It is based on a weighted least squares minimization of the intensity conservation constraint in each small spatial neighborhood

$$\min \sum W^2(\mathbf{x})\big[\nabla I(x, y)v + I_t(\mathbf{x}, t)\big]^2 \tag{5.47}$$

where W is a window that gives more importance to the constraints near the center of the chosen neighborhood. The solution of (5.47) is given by

$$v = \big[\mathbf{A}^{\mathrm{T}} W^2 \mathbf{A}\big]^{-1} \mathbf{A}^{\mathrm{T}} W \mathbf{b} \tag{5.48}$$

where

$$\mathbf{A} = \big[\nabla I(\mathbf{x}_i), \ldots, I(\mathbf{x}_n)\big]^{\mathrm{T}} \tag{5.49}$$

$$W = \mathrm{diag}\big[W(\mathbf{x}_i), \ldots, I(\mathbf{x}_n)\big] \tag{5.50}$$

$$\mathbf{b} = -\big[I_t(\mathbf{x}_1), \ldots, I_t(\mathbf{x}_n)\big]^{\mathrm{T}} \tag{5.51}$$

An interesting characteristic of the Lucas–Kanade algorithm is that it provides a measure of the estimation error, given that the matrix $[\mathbf{A}^{\mathrm{T}} W^2 \mathbf{A}]^{-1}$ is consistent with a covariance matrix. Therefore, unreliable estimates can be identified using the inverse eigenvalues of this matrix. However, this method is not suitable for dealing with displacements exceeding a pixel per frame, causing the estimation to fail.

Nevertheless, an extension of this method that computes the optical flow via a hierarchical coarse-to-fine process has been proposed. It is called the Lucas–Kanade pyramidal representation, and complements the original method with a pseudo-iterative scheme, allowing to compute the optical flow by propagating the flow in lower resolutions to larger resolutions [16].

In spite of all the advantages of the Lucas–Kanade algorithm, the optical flow estimated by this method represents the apparent movement of the objects in the scene, which may not correspond to the real object movement. In order to compute effectively the optical flow, the next points should be taken into account:

- The lightning has to stay constant over time. This hypothesis is the basis of the optical flow computation. Without it, (5.45) could not be written. If the illumination of the scene changes from one instant to another, objects may seem like being on movement, when in reality they are static over time. Indeed, according to the position of the light source, shadows are different, and even if the object is stationary, the imaging sensor can detect the movement of shadows.
- A rich-textured image is another important condition for accurate estimation of the optical flow. For a differential approach, like the Lucas–Kanade algorithm, the most contrasting textures have to be chosen in order to perform properly. A low-pass filter can be applied with the purpose of incrementing the image's contrast. Commonly, a Gaussian filter is used to create a sort of blur to the image. However, a simpler way of achieving the same effect consists of defocusing the lens, so the image becomes blurred. Nevertheless, filtering has to be performed carefully, since computing the optical flow requires good contrasts, but they should not be too discontinuous.
- A differential method alone does not solve what is called the *aperture problem*, which arises when using a small window to measure motion: one often sees only an edge, not a corner. But an edge alone is insufficient to determine exactly how (i.e., in what direction) the entire object is moving. In conclusion, one should observe more globally. However, this implies incrementing the computing time. A solution to this problem consists of eliminating the inconsistent points by means of the covariance matrix previously introduced.

Optical flow algorithms, like the Lucas–Kanade algorithm, are designed to work well with conventional CCD cameras. In the present studies, a conventional imaging sensor in combination with the pyramidal implementation of the Lucas–Kanade algorithm is used for computing optical flow.

5.4 Implementing an Imaging System for the Quad-Rotor UAV

Nowadays technological advances allow the development of vision systems well suited for being implemented on mini-UAVs. If imaging processing is intensive, the onboard imaging sensor can be connected to a wireless transmitter. This allows sending real-time video signal to a remote computer equipped with a video receiver.

Once images are received, a computer vision application processes the real-time video and extracts the required information. With the purpose of generating remote control commands for stabilizing the vehicle in flight, the imaging-based data can be used by a supervisory control program, running also in the remote computer. In a different approach, image-based measurements can be sent back directly to the aerial vehicle, in order to perform a data fusion and a control strategy onboard.

Recently, the robotics research community has increased its interest in the development of fully embedded video-processing systems for UAVs. However, the development of such system, for a mini-UAV like the quad-rotor, represents a very challenging task. Imaging processing is computationally expensive, therefore, it must be performed in a powerful processor. Unfortunately, the quad-rotor payload capacity is very limited, restricting considerably the available options for embedded computers. This situation motivated several research groups to develop their own prototypes and solutions for performing image processing.

This section presents first a brief discussion concerning deported and embedded vision systems. The following subject addresses several important characteristics that must be considering when implementing a monocular or a stereo imaging system onboard a quad-rotor UAV. Next, the development of a monocular and a stereo vision system for the quad-rotor platform are detailed. Both systems have been built with the purpose of evaluating the benefits and drawbacks of each approach in real-time experiments.

5.4.1 Deported and Embedded Systems

Nowadays, technological advances allows performing image processing onboard the flying robot or in a deported ground station PC. Generally, the deported solution is chosen when the computational tasks requires more resources than one could embed in the vehicle. Some vision algorithms such as optical flow estimation and SLAM schemes require working with dense fields (great number of points) and iterative approaches which are computational expensive. When imaging tasks are computationally expensive, the best solution is to use fast top-technology computers. However, these last ones are commonly heavy and consume too much power, which are two contradictory characteristics with respect to the design constraints of unmanned aerial vehicles. Nevertheless, the UAV research domain is taking advantage of the efforts that the electronic industry is doing due to the demand of developing small and powerful devices such as mobile phones, which are becoming smart devices. Thus, it is sure that, in a short time, all kinds of visual processing will be done onboard with very powerful, tiny and low power microcomputers. In the following sections, the different solutions associated with embedded and deported systems will be detailed, taking in consideration the quad-rotor design constraints previously explained.

Fig. 5.16 Analog camera
and video transmitter set-up

5.4.1.1 Deported Systems

Deported systems are based on wireless technology. Nowadays, there exist two different ways of realize this wireless connection: *wifi* and *radio frequency*. Radio frequency imaging system are basically conformed of an analog camera connected directly to a video transmitter. Figure 5.16 shows an analog camera–video transmitter set-up. This device sends the camera image through radio frequency data to the receiver placed on the ground, which in turns transfers the analog image to a frame grabber that performs the data conversion. Finally, the digitalized image is provided to the computer via USB. The communication between the video transmitter and receiver is executed at standard pre-defined working frequencies. There exist three standard frequencies allowed for video transmitting: 2.45 GHz, 1.2 GHz and 5.8 GHz. The most common frequency band used is the 2.45 GHz band, which is often very saturated by other wireless communications in certain areas. Generally, it is possible to choose compatible channels between the different communications present in the band. However, when a pair of devices are not built with the same frequency standard, it is impossible to suppress the interferences that could exist between them.

The principal disadvantage of radio frequency transmission is that it presents unwanted issues related to multipath signal transfer, which occur mostly in indoor environments where walls, furniture and other objects interfere with the signal's path occasioning signal rebounds. Multipath problems can cause blur and noisy images, which are very difficult to work with. For this reason, special attention is required in the way that the overall system handles noisy images since a control input resulting from a false image measurement can instantaneously cause the crash of the vehicle.

Visual systems working with wifi use a digital IP camera that establishes a TCP/IP network between the ground computer and the camera itself, see Fig. 5.17. The camera's image is transferred compressed in a digital format. Then, the ground computer has to decompress the image in order to be able of continuing with further processing. Generally the decompressing stage could take a considerably amount of time, causing a delay in the overall process of up to 1 ms. This situation results in

Fig. 5.17 Digital IP camera

a considerably lag between the instant of time when the image is taken and the moment when the respective control input arrives at the UAV. Obviously this situation is delicate when the stability of the aerial robot and the performance of the imaging system is executed in real-time. Nevertheless, the principal advantage of this digital transmission is that the data are not perturbed by other signals, which results in clear and noiseless images at the reception device. It is also worth to mention that deported systems are very useful when the weight constraint of the UAV is in its limits and when the computation process is very expensive.

5.4.1.2 Embedded Systems

Embedded visual system not only contains the optical sensor, but also has the computing resources to process the captured images. The principal advantage of such systems is that no lag is introduced in the imaging processing loop. However, depending on the capabilities of the onboard system processor, the working frequency could be more or less affected compared to the one working frequencies reached by an ordinary computer. Indeed, if the processor is not powerful enough to running the algorithm, the performance of the embedded systems will be worse that the one of the deported systems. Until recent years, the only embedded systems that could be carried by the quad-rotor were some dedicated optical flow sensors and the CMU-Cam. For example, the VLSI sensor developed presented in [10] was patented and first commercialized under the name of LadyBug sensor, see Fig. 5.18(a), then it was upgraded to the Mantis sensor, see Fig. 5.18(b), and finally to the Arz sensor Fig. 5.18(c).

The CMUCam3 is another interesting visual embedded system first developed by the Carnegie Mellon University. Such system is programmable, allowing the users to adapt their own algorithms in order to be executed by the processor available on the system. However, this is a low-cost solution where most of the complex algorithms

Fig. 5.18 Centeye optical flow sensors: (**a**) LadyBug; (**b**) Mantis; (**c**) Arz

(a)   (b)   (c)
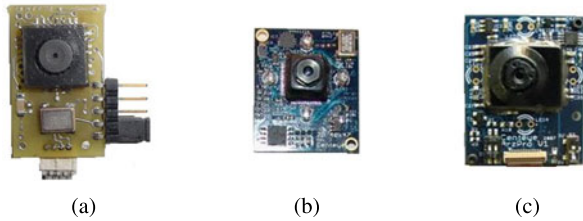
Fig. 5.19 Gumstix Overo Fire

already adapted for this system run a low rates, which is not appropriate for aerial vision-based navigation.

Nowadays, with the size and weight reductions of high-performance processors, computer-on-module systems are taking an important place in the market. The computer vision field is taking advantage of those developments to finally implement very complex and accurate algorithms in small size devices. The Gumstix Overo Fire, see Fig. 5.19, belongs to the family of computer-on-module systems. With a tiny ARM Cortex-A8 OMAP3530 processor, a TI C64x DSP and the POWERVR SGX for graphics acceleration, the Overo Fire can handle time-consuming algorithms that require considerably high computing resources. Different teams working on UAVs have started to developed visual systems based on this device with very promising results. However, this device still has a limited range of applications.

In addition to those systems, Ascending Technologies have designed a computer-on-module specially adapted for visual applications on UAVs. The system is conformed by a CoreExpress board from Lippert Embedded, see Fig. 5.20(a). Since the processor is a X86 architecture, standard operating systems and drivers for devices like cameras, laser range finders or WiFi could be used. The development of such powerful computer-on-module has attracted the attention to other systems even more powerful. The PIXHAWK team from the Computer Vision and Geometry Lab at the ETH Zurich have developed a computer-on-module baseboard to carry the microETXexpress-PC from Kontron, see Fig. 5.20(b). This system is a very powerful computer-on-module, consisting of an Intel Core 2 Duo SL9400 running at 1.86 GHz with 6 MB L2 Cache, a 2 GB DDR3 memory chipset, and the Intel GMA X4500 integrated graphics. This is a very complete system that can be compared to a netbook computer. However, it weighs 235 grams, which could be a serious problem when carried by a quad-rotor vehicle. The implementation of such system
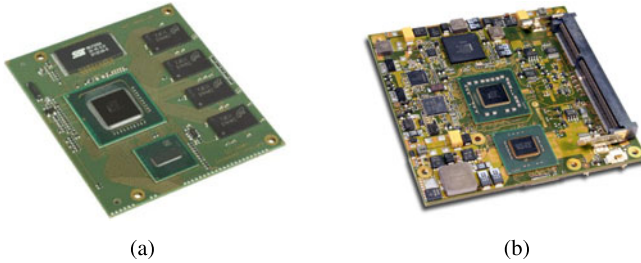
Fig. 5.20 Computer-on-module sensors: (**a**) Lippert Embedded CoreExpress board; (**b**) Kontron microETXexpress-PC

on the overall design of the aerial vehicle requires the optimization of the rest of the vehicle, in order to save weight for the vision computer.

5.4.2 Challenges when Using Monocular and Stereo Imaging Systems

With the purpose of estimating relative positioning and velocity where other sensing technologies cannot, monocular and stereo vision systems are being installed onboard UAVs. A discussion concerning important characteristics of each one of those systems is given, as well as some challenges, advantages and disadvantages, encountered during their implementation.

Monocular Imaging System   From a set of consecutive images, motion as well as 3-dimensional structure can be estimated by a single camera. First of all, the essential matrix E or fundamental matrix F describing the relative motion between two views has to be estimated. This can be achieved by using feature correspondences $\mathbf{x}_1 - \mathbf{x}_2$ projected from a set of 3-dimensional structures $\mathbf{X}_i$ into these views. Decomposing this matrix yields four possible solutions for rotation and translation, respectively. Only one of the four solutions is physically possible, which can be identified by reconstructing a single 3-dimensional point, and verifying if it is located in front w.r.t. the camera frame. Once the correct combination of rotation and translation is known, the 3-dimensional structure can be reconstructed from the image correspondences by the method of triangulation. The rotation estimation is unique and causes no problems, however, translation as well as the depth of the structure can only be recovered up to a scalar factor which results from the fact that the essential matrix only has a rank of 2. Hence, only the translational direction of the camera can be determined, not the magnitude.

In order to recover the essential matrix and 3-dimensional structure robustly w.r.t. to unavoidable noise, a sufficiently large baseline between two views is necessary, as well as two distinct vantage points. Unfortunately, when working with a quad-rotor UAV both previous requirements are difficult to meet, since this kind of vehicles

normally perform small movements between consecutive images. In addition, their movement is mostly performed in the camera's direction of view. The first problem can be solved by skipping images until the baseline between the evaluated par of images is large enough. Unfortunately, this approach does not meet the UAVs control requirement of having fast position and velocity estimates.

When using a monocular system, the 3-dimensional structure can only be recovered as long as the vehicle is moving. Since it is desirably that the quad-rotor navigates slowly while performing missions, 3-dimensional reconstruction becomes difficult, making very problematic some elementary tasks like obstacle avoidance.

To overcome the scale ambiguity issue when using monocular vision systems, either some previous scene knowledge is necessary, or information from other sensors (range finder, IMU) have to be taken into account. By placing artificial landmarks of well known dimensions in the environment where the UAV performs, the scale ambiguity problem can be solved effectively. However, since UAVs are meant to fly into unknown environments, previous knowledge of the scene is usually not available. On the other hand, incorporating additional sensors yields also additional problems, like having different measurement rates or inaccurate calibration from these sensors w.r.t. the imaging sensor.

Stereo Imaging System   A stereo vision system allows to reconstruct 3-dimensional structure as well as motion directly, independently of the motion itself. Placing the pair of cameras in a considerable distance from one to the other, a stereo rig enforces a sufficiently large baseline between both views. In addition, it also allows to reconstruct the 3-dimensional feature position in a single time-step. In addition, stereo vision techniques can deal with the degenerate configuration in which all 3-dimensional features considered lie on a plane.

The stereo vision systems have also some disadvantages that are worth to mention. A stereo vision algorithm is computationally more intensive and demanding than monocular vision algorithms. Having two cameras, all image processing tasks like undistortion, rectification and filtering have to be done twice. In addition, the feature's correspondences have to be performed between consecutive images, but also between the left and right images. Furthermore, some hardware issues must be solved, like image processing bus overload, or effective synchronization of the cameras. If the stereo cameras are not synchronized, and the vehicle moves between the acquisition of the stereo pair, the relative pose of the cameras will change and, as consequence, the extrinsic calibration will not fit any more. This penalizes the 3-dimensional structure reconstruction, leading to inaccurate pose estimates. Considering the fast dynamics of the quad-rotor, synchronized cameras are essential for performing stereo imaging studies.

Implementing a stereo imaging system also implies that more components need to be installed onboard, which increases the weight that the UAV must lift. In order to overcome this situation, the quad-rotor design must be carefully conducted. As a conclusion, light frame and onboard components must be chosen, as well as powerful motors, aiming at increasing the UAV payload capacity.
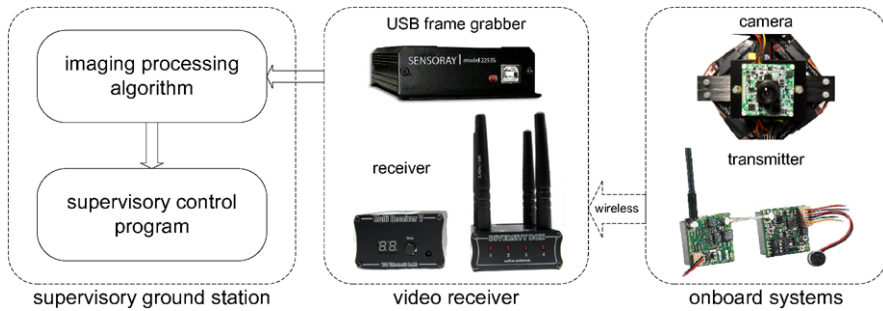
Fig. 5.21 Schema of the monocular vision system

5.4.3 Monocular Imaging System Implementation

The quad-rotor's monocular imaging system was developed with the purpose of enabling the vehicle for performing vision-based tasks. The imaging sensor is a *CTDM-5351* camera from SONY, with a resolution of 640×480 pixels, installed in the lower part of the vehicle, placed pointing downwards. The *CTDM-5351* camera has been chosen, since it offers a considerably high quality image within well and poorly illuminated areas. The analog output of the camera is connected directly to a wireless *Micro PLL Transmitter* and a *Micro Booster* of 200 mW. By combining these two components, the transmission power can be improved up to 20 times, ensuring a good quality video during the UAV's missions. With the purpose of reducing possible disturbances in the transmitted images, the video signal is recovered by means of a 4-antenna *Diversity System Receiver*. Incoming signals are received by each one of the four antennas, so that the receiver can evaluate which antenna is providing the most suitable signal. Once detected, the system switches to this antenna automatically. The receiver outputs the video signal via a composite connector to a *Sensoray 2255* USB Frame Grabber, which is specially designed to fast video acquisition. The overall system, composed by transmitter, receiver, and frame grabber, was chosen with the purpose of reducing possible time delays, and to ensure a high quality real-time video.

The frame grabber is connected to the supervisory ground station via a USB port. A C-coded imaging application, based on OpenCV library functions [20], performs image processing for estimating the vehicle's states. OpenCV [64] is a functions library, containing a series of low-overhead, high-performance operations, that can be used to perform fast computing algorithms on images. An overall scheme of the monocular system components is shown Fig. 5.21.

The information extracted by the imaging application is placed on a fixed memory segment that is shared with the supervisory control application. With this method, the supervisory control application receives the required data for performing vision-based tasks.
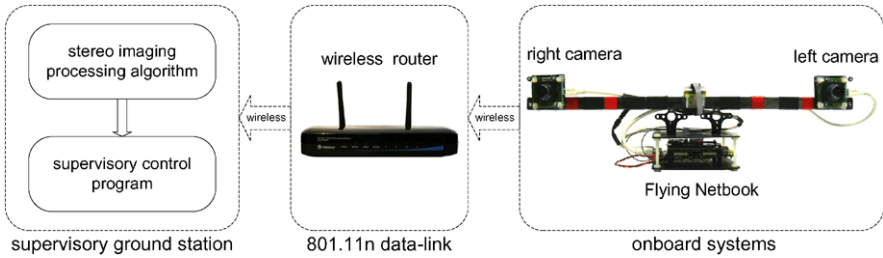
Fig. 5.22   Schema of the stereo imaging system

5.4.4 Stereo Imaging System Implementation

A schema showing the quad-rotor's stereo imaging system can be seen in Fig. 5.22. This arrangement is designed to be installed on the top of the helicopter, positioned facing forward. Two monochromatic *uEye UI-1226LE-M-G* cameras from *IDS GmbH*, having a resolution of 376×240 pixels and a field of view of $75°$ have been chosen as imaging sensors. The cameras are mounted on a carbon rig, separated one from another by a distance of 35 cm. As can be seen in the corresponding image, an additional IMU has been installed in a centered position between both cameras. The purpose of this inertial sensor will be explained later, in Chap. 7. The stereo imaging system has been accurately calibrated as shown previously in Sect. 5.2.2, which allows the evaluation of the epipolar constraint from (5.23).

The cameras operate in an external trigger mode, therefore, every image acquisition has to be requested by sending a hardware trigger signal to each camera. *uEye* cameras are equipped with hardware input/output ports, one of which has been programmed as hardware trigger input. The left camera is enabled as *master*, and one of its input/output ports is configured as an output that is triggered by software. This output is connected to the left as well as to the right camera's trigger input, therefore, both cameras are waiting for the hardware trigger to perform image acquisition.

The cameras are connected via mini-USB ports to an embedded Flying Netbook board, developed by *Ascending Technologies GmbH* [8]. This mini computer counts with an Intel Atom 1.6 GHz processor, which gives enough power for executing a C-coded program based on OpenCV functions, which deals with the next group of tasks. The first step of the program consists of generating the trigger signal required for synchronizing the *uEye* cameras. Once the cameras receive the trigger, a synchronized images pair is captured. The program proceeds to recuperate both images, next, it uses OpenCV functions to compress them in jpeg format. Both images are then stored in a data buffer, which is sent via a 801.11n wireless data-link to the supervisory ground station PC. Communication between the *Flying Netbook* and the supervisory ground station is achieved by using a *TEW-652BRP Wireless Nspeed Broadcast Router* from *TRENDnet*.

Once the data buffer is received, a C-coded application based on OpenCV functions decompresses the images and performs the required imaging processing over the stereo pair. This step provides accurate measurements of the helicopter (x, y, z)
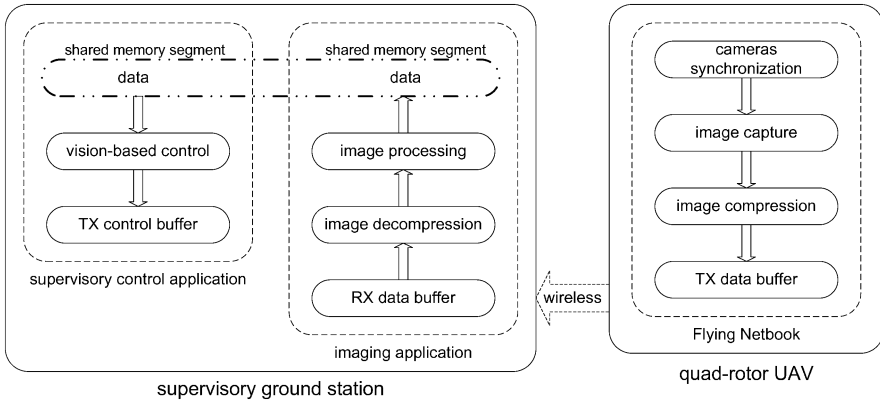
Fig. 5.23 Schema of the stereo imaging process and the vision-based control computation

3-dimensional position relative to its surrounding environment. Such data are placed on a fixed memory segment that is shared with the supervisory control application. With this method, the supervisory control application receives the required data for performing vision-based tasks. A schematic showing the stereo imaging process and the vision-based control computation is shown in Fig. 5.23.

5.5 Concluding Remarks

This chapter provided fundamental background behind the implementation of imaging sensors for estimating relative position and velocity. With the purpose of estimating the internal parameters of a camera, the calibration of a monocular system has been addressed. In addition, a stereo imaging system has been calibrated, which allowed the reconstruction of a 3-dimensional point from its corresponding 2-dimensional projections. Both calibration methods were based on the *Camera Calibration Toolbox for Matlab*. For the computation of relative translational speed using an imaging sensor, the concept of optical flow and a method for its computation have been detailed.

Some important points that must be considered when implementing a monocular or a stereo imaging system onboard a quad-rotor UAV have been analyzed and discussed. This allowed the identification of some benefits and drawbacks inherent to each approach, and motivated the development of both systems for their evaluation in real-time experiments. The components conforming the monocular system, as well as the software architecture allowing the estimation of the data required for performing vision-based tasks were presented. A similar explanation is given for the stereo imaging system.

The following sections of this book show the performance of both systems for estimating the states of a quad-rotor UAV, which allows controlling the flight of the vehicle during flight experiments.

Chapter 6
Vision-Based Control of a Quad-Rotor UAV

This chapter presents two different vision-based strategies for stabilizing a quad-rotor UAV during flight. Theoretical and practical aspects are detailed, as well as the physical set-ups developed, allowing the realization of real-time experimental applications.

The chapter is divided as follows. Section 6.1 presents a vision-based strategy for stabilizing the quad-rotor during flight. This technique is based on a homography estimation technique and an optical flow computation. Next, a comparison of three control strategies is addressed in Sect. 6.2, with the purpose of identifying the most effective approach for stabilizing the vehicle when using visual feedback. In Sect. 6.3 the vision system is implemented for allowing altitude control, with the objective of stabilizing the 3-dimensional position and regulating the velocity of the vehicle using optical flow. Finally, some concluding remarks are presented in Sect. 6.4.

6.1 Position Stabilization Using Vision

6.1.1 Introduction

With the purpose of enhancing the autonomy level of the "Cross-Flyer" UAV presented in Sect. 3.4.1, a vision-based position control approach is proposed. The helicopter's (x, y, z) positions and $(\dot{x}, \dot{y}, \dot{z})$ velocities are estimated with respect to a landing pad on the ground, using the monocular vision system presented in Sect. 5.4.3. The developed technique allows measuring the states, which are difficult to obtain from conventional navigation systems, for example GPS, when performing in urban environments or indoors.

In the proposed approach, the 3-dimensional position estimation is based on the computation of homographies, while translational velocity is obtained based on an optical flow computation. The states estimated are applied in a full state feedback controller, with the purpose of generating control input for stabilizing the quad-rotor

Fig. 6.1 Vision-based
position stabilization schema
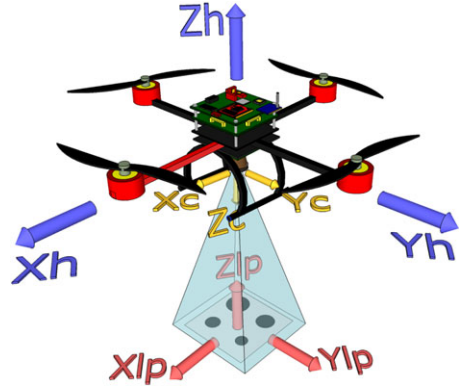
(x, y, z) position during flight. The effectiveness of the proposed method has been verified under real-time experiments. Some graphics representing the behavior of the helicopter are shown to illustrate the performance of the UAV during flight.

6.1.2 Visual System Set-up

Controlling the 3-dimensional position of a UAV depends on the knowledge of the (x, y, z) vehicle coordinates and $(\dot{x}, \dot{y}, \dot{z})$ translational velocities with respect to a well-known reference frame. Such values are required data for the controller in order to generate the control input to stabilize the aircraft over a desired location. To fulfill this situation, the vision system presented in Sect. 5.4.3 has been implemented, in order to provide the required position and velocities information. The complete system proposed consists of the calibrated camera onboard the UAV, a landing pad or artificial marker placed on ground, an imaging processing algorithm running on a supervisory ground station PC, and the wireless link between the helicopter and the supervisory ground station. Figure 6.1 shows the proposed system which can be described as:

- **Quad-rotor UAV:** with a body fixed frame (X_h, Y_h, Z_h), assumed to be at its center of gravity. Z_h represents the yaw axis, and pointing upwards. X_h and Y_h are the roll and pitch axis, respectively.
- **Strapdown camera:** pointing downwards, with a reference frame (X_c, Y_c, Z_c). When moving, the camera surveys the scene passing below the quad-rotor. Since $X_c - Y_c$ and $X_h - Y_h$ are considered as parallel planes, then, the visual information collected by the camera can be used to stabilize the vehicle.
- **Landing pad:** artificial landmark of known dimensions, formed by four circles of known coordinates, painted on high contrast background and placed underneath the rotorcraft. The coordinates frame (X_{lp}, Y_{lp}, Z_{lp}) represents the inertial reference frame.

The planes formed by $(X_h - Y_h)$ and $(X_{lp} - Y_{lp})$ are considered to be parallel because it is assumed that the rotorcraft is in hover flight over the landing pad. Running in the supervisory ground station, an algorithm for imaging processing provides an estimate of the helicopter altitude, position in the (X, Y) plane, and translational velocities. As mentioned earlier, the 3-dimensional position information is deduced by an homography estimation technique applied to the image of the landing pad, while the translational velocities are estimated by means of optical flow computation. All information sensed by the camera is related to the landing pad image. It will be shown that this information is rich enough to stabilize the 3-dimensional position of the vehicle. Let us explain first the procedure performed to compute the vehicle's positions and velocities.

6.1.3 Vision-Based Position Estimation

This section describes the imaging algorithms developed with the purpose of estimating the vehicle's states required for stabilizing its position during flight. First, a homography estimation technique for computing the 3-dimensional position is described, followed by a technique for deriving translational velocities. Finally, a method for estimating the homography when the detection of the landing pad fails is presented.

6.1.3.1 Computing the 3-dimensional Position

In order to estimate the UAV position relative to the landing pad, the extrinsic parameters of the camera are computed at every image frame. This is achieved by implementing a homography estimation technique, which provides the (x, y, z) position and (ψ, θ, ϕ) orientation of the camera with respect to the artificial landmark in the image scene. The action of the homography can be expressed as [20]

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sK \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.1}$$

where $[x \ y \ 1]^T$ represents the landing pad position in the camera image, s is a known scale factor, $K \in \mathbb{R}^{3 \times 3}$ represents the intrinsics parameters camera matrix (previously found in (5.19)), $R = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3] \in \mathbb{R}^{3 \times 3}$ are the extrinsic rotation parameters, $\mathbf{T} \in \mathbb{R}^{3 \times 1}$ is the extrinsic translation parameters vector and $[X \ Y \ Z \ 1]^T$ is the real landing pad position.

Without loss of generality, the landing pad plane is defined so that $Z = 0$. This is done because, if one also breaks up the rotation matrix into three 3×1 columns (i.e., $R = [r_1 \ r_2 \ r_3]$), then one of those columns is not needed. Therefore, the homography matrix H can be expressed as $H = sK[r_1 \ r_2 \ \mathbf{T}]$. The homography matrix

H is divided in two parts: the physical transformation (which locates the observed object plane) and the projection (the camera intrinsic matrix).

Rotation R is described by three angles and translation $\mathbf{T}$ is defined by three offsets; hence there are six unknowns for each view. The known planar object (the artificial landmark) provides eight equations, that is, the mapping of a rectangle into a quadrilateral can be described by four (x, y) image points. For every instant, when the aerial vehicle is in hovering, it is possible to compute the homography matrix H using the a priori knowledge of the position of the four centroids of the circles [44]. Using this estimated transformation matrix and the intrinsic camera matrix previously identified by an off-line calibration based on the method in [17], it is possible to calculate the camera extrinsic parameters, and therefore one has the vehicle's (x, y, z) position with respect to the landing pad on the ground.

Each one of the circles are detected in the image using an OpenCV function, next, they are classified according to the magnitude of its radius, allowing a correct identification of the landing pad orientation. The first circle corresponds to the upper left circle (or circle with smallest radius), continuing with the upper right circle as the second in the list. The lower left circle comes next according to its radius, and finally, the lower right circle is identified as the circle whose radius magnitude is bigger. Figure 6.2 shows an image of the landing pad, as viewed from the camera, where the four circles perimeters are highlighted with different colors according to the magnitude of its radius.

An erroneous detection of the landing pad circles must be discarded, since it will provide an erroneous position estimation. With this purpose, the parallelism of the lines mapped from the landing pad must be verified. Figure 6.1 shows that the four circles of the landing pad are positioned forming the corners of a rectangle. The line between the two upper corners and the line joining the two lower corners must satisfy a parallelism constrain. The same restriction is checked for the line joining the two left corners and the line between the two right corners. Parallelism verification is based on the slope of a line equation:

$$m_* = \frac{y_f - y_i}{x_f - x_i} \tag{6.2}$$

where i and f stand for initial and final coordinates, respectively. Thus, the slope m_{up} of the upper line must be almost equal to the slope m_{lo} of the lower line, while the slope m_{le} of the left line must be almost equal to the slope m_{ri} of the right line:

$$|m_{\mathrm{up}} - m_{\mathrm{lo}}| < \varepsilon; \qquad |m_{\mathrm{le}} - m_{\mathrm{ri}}| < \varepsilon \tag{6.3}$$

where ε stands for a constrain helping to determine the lines parallelism. Every detection of the four circles, validating the two previous verifications, ensures that a good planar homography could be estimated, resulting in a good computation of the camera extrinsic parameters. The four lines between circles, defining the sides of the rectangle, are highlighted in Fig. 6.2.

Fig. 6.2 Image processing
for homography estimation:
Detection of the four circles
and parallel lines verification

6.1.3.2 Translational Velocities

An optical flow computation procedure is applied to compute the $(\dot{x}, \dot{y}, \dot{z})$ translational velocities of the aerial vehicle with respect to the landing pad. In order to compute optical flow, the Lucas–Kanade pyramidal algorithm [16] has been implemented in combination with a feature-detecting algorithm. This approach provides an accurate estimation of the motion field since it does not take into account the non landing pad areas, where the motion field cannot be accurately determined.

Let us consider the camera moving with respect to a rigid scene. The velocities and rotation rates of the camera in the inertial frame are expressed by (V_x, V_y, V_z) and (w_x, w_y, w_z), respectively. To accurately estimate the pseudo-speeds of the engine, let us define a tracking zone surrounding the landing pad, in a way that the centroid of the zone and the center of the landing pad coincide. The most representative features over the zone are detected using an OpenCV function devoted to such task. These features, usually the circle perimeters, are selected as *features to track for*. A tracking process based on OpenCV-based optical flow computation is performed over the entire image, allowing measuring the displacements of the tracked features.

Thus, the mean of the optical flow computed on all those points can be expressed as a function of the camera movement as follows:

$$\bar{\text{OF}}_x = \bar{V}_{\text{OF}_x} + K_x \bar{V}_{\text{OF}_z} + \bar{R}_{\text{OF}_x} \tag{6.4}$$

$$\bar{\text{OF}}_y = \bar{V}_{\text{OF}_y} + K_y \bar{V}_{\text{OF}_z} + \bar{R}_{\text{OF}_y} \tag{6.5}$$

Using the results presented in [68], the rotational optical flow is compensated and the pseudo-speeds $(\bar{V}_{\text{OF}_x}, \bar{V}_{\text{OF}_y}, \bar{V}_{\text{OF}_z})$ are deduced. Since the camera system and the helicopter share the same movements, it can be said that the deduced pseudo-velocities depend of the rotorcraft movement. Indeed, the camera is mounted on-board the quad-rotor and fixed in a way it has no freedom degree. Thus, we may write

$$\bar{V}_{\text{OF}_x} = -\frac{f\dot{x}}{z} \tag{6.6}$$

$$\bar{V}_{\text{OF}_y} = -\frac{f\dot{y}}{z} \tag{6.7}$$

$$\bar{V}_{\text{OF}_z} = \frac{\dot{z}}{z} \tag{6.8}$$

where $(\dot{x}, \dot{y}, \dot{z})$ is the speed vector of the rotorcraft center of gravity and z is the altitude. Thus, from these three equations the proposed optical flow vision system allows speed estimation of the rotorcraft up to a scale factor, when flying at constant altitude. Those estimations can be used to control the translational velocities of the rotorcraft.

6.1.3.3 Prediction of the Landing Pad Position

Changes of illumination between one frame to another as well as occlusions can lead to a poor performance of imaging processing algorithms. Due to this, if the landing pad is not successfully detected in the current image, the vision algorithm will fail. To overcome such situation, optical flow measurements are applied to estimate the position of the four circles centroids in the current image as

$$\rho_x^k = \rho_x^{k-1} + \Delta_T \bar{V}_{\text{OF}_x} \tag{6.9}$$
$$\rho_y^k = \rho_y^{k-1} + \Delta_T \bar{V}_{\text{OF}_y} \tag{6.10}$$

where (ρ_x^k, ρ_y^k) represents the circle's centroid position at time instant k, and Δ_T is the working frequency of the algorithm. From this is evident the benefit obtained from applying optical flow for computing the translational velocities. The centroid positions estimated from (6.9)–(6.10) allow to compute the homography at each frame in case that the detection of the landing pad fails.

6.1.4 Control Strategy

Since there exists a wireless communication between the helicopter and the supervisory ground station, the hierarchical controller presented in Sect. 3.5 can be implemented to perform deported vision-based stabilization tasks. The control scheme for the overall cascaded system is then formed by:

- **Position control:** a high-level control running in the supervisory ground station.
- **Attitude control:** a low-level control, performed by the autopilot embedded in the vehicle.

The imaging algorithm described in Sect. 6.1.3 provides the vehicle's states required in the control strategy. Such information is shared between the supervisory control program and the imaging application by means of a shared memory, as explained previously in Sect. 3.3.

6.1.4.1 Altitude and Yaw Control

The control of the vertical position (2.32) can be obtained by using the following control input:

$$u = (-k_{vz}\dot{z} - k_{pz}e_z + mg)\frac{1}{\cos\theta\cos\phi} \qquad (6.11)$$

with $e_z = z - z_d$ as the z error position and z_d as the desired altitude. k_{pz} and k_{vz} are positive constants. Thus, for the altitude dynamics, r_1 is a PD controller. In the case of the yaw angular position (2.33), one can apply

$$\tilde{\tau}_\psi = -k_{v\psi}\dot{\psi} - k_{p\psi}e_\psi \qquad (6.12)$$

where $e_\psi = \psi - \psi_d$ denotes the yaw error, ψ_d represents the desired yaw angle, $k_{p\psi}$ and $k_{v\psi}$ denote the positive constants of a PD controller. Indeed, introducing (6.11) and (6.12) into the set of equations (2.30)–(2.35) and provided that $\cos\theta\cos\phi \neq 0$, one has

$$m\ddot{x} = (-k_{vz}\dot{z} - k_{pz}e_z + mg)\left(\frac{\sin\psi\tan\phi}{\cos\theta} + \cos\psi\tan\theta\right) \qquad (6.13)$$

$$m\ddot{y} = (-k_{vz}\dot{z} - k_{pz}e_z + mg)\left(\sin\psi\tan\theta - \frac{\cos\psi\tan\phi}{\cos\theta}\right) \qquad (6.14)$$

$$m\ddot{z} = -k_{vz}\dot{z} - k_{pz}e_z \qquad (6.15)$$

$$\ddot{\psi} = -k_{v\psi}\dot{\psi} - k_{p\psi}e_\psi \qquad (6.16)$$

The control parameters $k_{p\psi}$, $k_{v\psi}$, k_{pz}, and k_{vz} should be carefully chosen to ensure a stable well-damped response in the vertical and yaw axes [27]. From (6.15) and (6.16) it follows that $\psi \to \psi_d$ and $z \to z_d$. From (6.11)–(6.12) and (6.15)–(6.16) $r_1 \to 0$ and $\psi \to \psi_d$. For a time T large enough, e_z and e_ψ are arbitrarily small, therefore, (6.13) and (6.14) reduce to

$$\ddot{x} = g\tan\theta \qquad (6.17)$$

$$\ddot{y} = -g\frac{\tan\phi}{\cos\theta} \qquad (6.18)$$

6.1.4.2 Control of Forward Position and Pitch Angle

Consider the subsystem given by (2.34) and (6.17). Let us impose a very small upper bound on $|\theta|$ in such a way that the difference $\tan(\theta) - \theta$ is arbitrarily small ($\theta \approx \tan(\theta)$). Therefore, the subsystem (2.34) and (6.17) becomes the following linearized system:

$$\ddot{x} = g\theta \qquad (6.19)$$

$$\ddot{\theta} = \tilde{\tau}_\theta \qquad (6.20)$$

The longitudinal subsystem focuses on the x–θ stabilization. The goal is to control the pitch angle θ and the longitudinal position x to desired references. Consider

the linearized system (6.19)–(6.20), the longitudinal position of the vehicle can be described as a chain of integrators,

$$\dot{x}_1 = x_2 \tag{6.21}$$

$$\dot{x}_2 = x_3 \tag{6.22}$$

$$\dot{x}_3 = x_4 \tag{6.23}$$

$$\dot{x}_4 = \tilde{\tau}_\theta \tag{6.24}$$

where x_1 is the error between the desired and the actual position, $x_3 = \theta$ and $x_4 = \dot{\theta}$. The hierarchical control is constructed by separating, in the previous subsystem, the rotational dynamics from the translational dynamics (low-level control and position control, respectively). Considering x_3 as the control input for the translational dynamics, and applying a backstepping change of variables, one has

$$\dot{x}_1 = x_2 \tag{6.25}$$

$$\dot{x}_2 = x_3^{\text{ref}} + \tilde{x}_3 \tag{6.26}$$

$$\dot{\tilde{x}}_3 = \tilde{x}_4 \tag{6.27}$$

$$\dot{\tilde{x}}_4 = \ddot{x}_2^{\text{ref}} + \tilde{\tau}_\theta \tag{6.28}$$

Here $x_3^{\text{ref}} = \theta_{\text{ref}}$, $\tilde{x}_3 = \theta - \theta_{\text{ref}}$, and $\tilde{x}_4 = \dot{\theta} - \dot{\theta}_{\text{ref}}$. θ_{ref} is the reference angle chosen to control the longitudinal displacement. Notice that setting the reference pitch angle to zero equals to stop the latter displacement. Since the low-level (rotational) control runs at highest rate than the navigation control, the reference angle dynamics can be ignored, then ($\dot{\theta}_{\text{ref}} = \ddot{\theta}_{\text{ref}} = 0$). Indeed, the dynamics of θ will converge faster than the position control. Then, one has

$$\dot{x}_1 = x_2 \tag{6.29}$$

$$\dot{x}_2 = x_3^{\text{ref}} + \tilde{x}_3 \tag{6.30}$$

$$\dot{\tilde{x}}_3 = x_4 \tag{6.31}$$

$$\dot{x}_4 = \tilde{\tau}_\theta \tag{6.32}$$

The longitudinal position is stabilized by defining the reference pitch angle as a function of the longitudinal position and velocity

$$\theta_{\text{ref}} = V_x(x, \dot{x}) = -k_d^x \dot{x} - k_p^x (x - x_d) \tag{6.33}$$

which can be defined as PD controller. The pitch reference angle is used in the rotational control as

$$\tilde{\tau}_\theta = -k_v^\theta x_4 - k_p^\theta \tilde{x}_3 \tag{6.34}$$

$$\tilde{\tau}_\theta = -k_v^\theta \dot{\theta} - k_p^\theta (\theta - \theta_{\text{ref}}) \tag{6.35}$$

The controller gains k_v and k_p should be chosen appropriately so that the polynomial $s^2 + k_v s + k_p$ is Hurwitz.

6.1.4.3 Control of Lateral Position and Roll Angle

Consider the subsystem given by (2.35) and (6.18). Let us impose a very small upper bound on $|\phi|$ in such a way that the difference $\tan(\phi) - \phi$ is arbitrarily small ($\phi \approx \tan(\phi)$). Therefore, the subsystem (2.35) and (6.18) becomes

$$\ddot{y} = -g\phi \tag{6.36}$$

$$\ddot{\phi} = \tilde{\tau}_\phi \tag{6.37}$$

The translational subsystem focuses on the y–ϕ stabilization. Consider the linearized system equations (6.36)–(6.37), the translational position of the vehicle can be described as a chain of integrators

$$\dot{y}_1 = y_2 \tag{6.38}$$

$$\dot{y}_1 = y_2 \tag{6.39}$$

$$\dot{y}_2 = y_3 \tag{6.40}$$

$$\dot{y}_3 = \tilde{\tau}_\phi \tag{6.41}$$

Considering ϕ as the control input for the translational dynamics, then the chain of four integrators can be separated into two subsystems composed of two integrators for both of them. The implementation of a backstepping change of variables yields

$$\dot{y}_1 = y_2 \tag{6.42}$$

$$\dot{y}_2 = y_3^{\text{ref}} + \tilde{y}_3 \tag{6.43}$$

$$\dot{\tilde{y}}_3 = y_4 \tag{6.44}$$

$$\dot{y}_4 = \tilde{\tau}_\phi \tag{6.45}$$

where $y_3^{\text{ref}} = \phi_{\text{ref}}$. Using high gains in the low-level attitude control, the dynamics of the reference angle can be ignored. The lateral position is then stabilized by defining the reference roll angle as a function of the lateral position and velocity,

$$\phi_{\text{ref}} = V_y(y, \dot{y}) = -k_d^y \dot{y} - k_p^y(y - y_d) \tag{6.46}$$

which can be defined as PD controller. The roll reference angle is used in the rotational control as

$$\tilde{\tau}_\phi = -k_v^\phi y_4 - k_p^\phi \tilde{y}_3 \tag{6.47}$$

$$\tilde{\tau}_\phi = -k_v^\phi \dot{\phi} - k_p^\phi(\phi - \phi_{\text{ref}}) \tag{6.48}$$

The controller gains k_v and k_p should be chosen appropriately so that the polynomial $s^2 + k_v s + k_p$ is Hurwitz.

Fig. 6.3 The four-rotor
aircraft experimental platform
with the camera pointing
downwards

Fig. 6.4 The experimental
system configuration:
"Cross-Flyer" UAV and
supervisory ground station
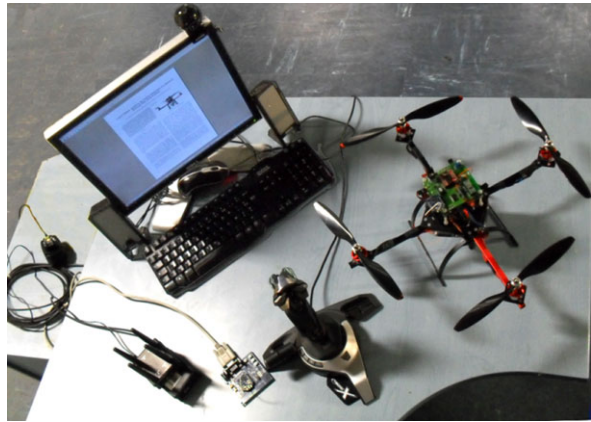
6.1.5 Experimental System Configuration

The imaging algorithms presented in Sect. 6.1.3 and the hierarchical controller presented in Sect. 6.1.4 have been tested over a system composed by the "Cross-Flyer" UAV platform, a supervisory ground station, and a wireless video link (onboard transmitter, on ground receiver).

The "Cross-Flyer" UAV with the downward looking camera is shown in Fig. 6.3. The supervisory ground station shown in Fig. 3.2 was used for these studies. It consists of a desktop PC, the joystick and the XBee ZB ZigBee PRO radio modem, and the 4-antenna *Diversity System Receiver*. The ground station runs a supervisory control application allowing a user to send information to the helicopter and to chose between a manual control or an autonomous vision-based position hold. The supervisory ground station receives and saves data sent by the vehicle in order to debug and analyze the flight experiments. The control feedback between the supervisory ground station and the helicopter is performed at 15 Hz. The complete experimental system can be seen in Fig. 6.4.
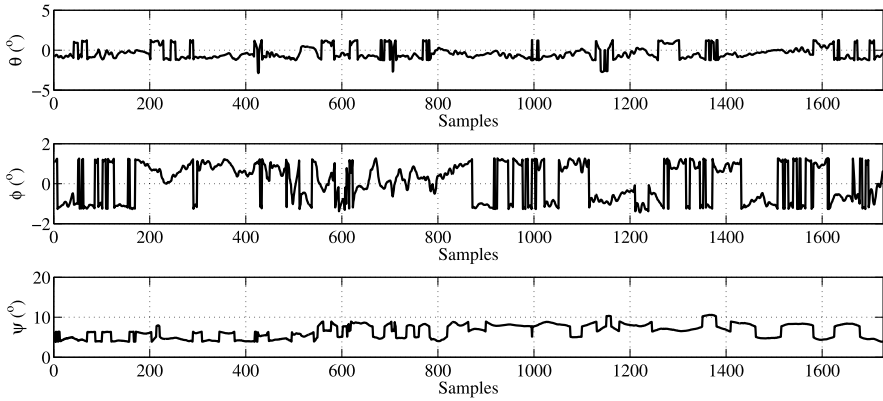
Fig. 6.5 Euler angles, experimental results

6.1.6 Experimental Applications

With the purpose of validating the vision-based control strategy proposed, a set of experiments were conducted. The goal consists of stabilizing the 3-dimensional position of the quad-rotor during flight, with respect to the landing pad placed on the ground. During the tests, the vision system provides the position and velocity feedback, while the embedded inertial electronic system provides the attitude data. The experiment's procedure is as follows.

First, the UAV is positioned exactly on top of the landing pad. Next, the operator uses the joystick in the supervisory ground station to define the current vehicle's position as the desired (x_d, y_d) position reference. The desired altitude z_d is always fixed at 150 cm, and the desired yaw angle ψ is fixed at 5° degrees. Using such values, the control strategy deals with stabilizing the vehicle during experiments. The control parameters used are: $k_{pz} = 0.68$, $k_{vz} = 1.6$, $k_{p\psi} = 38$, $k_{v\psi} = 1350$, $k_{px} = 1$, $k_{vx} = 2$, $k_p^\theta = 38$, $k_v^\theta = 1350$, $k_{py} = 1$, $k_{vy} = 2$, $k_p^\phi = 38$, $k_v^\phi = 1400$. As a reminder, the attitude stabilization control system is always running at higher frequency in order to guarantee that the Euler angles are close to zero (hover flight).

The set of Figs. 6.5, 6.6, and 6.7 show the results obtained from the experiment. In those figures, each 10 samples are equivalent to 1 second. Notice that the pitch and roll angles remain in the interval $(-1.5°, 1.5°)$ degrees. Therefore, it can be concluded that the position control adds only small changes in the attitude of the rotorcraft for bringing the position to the desired one. This is an important property because the position controller runs at a lower rate compared to the attitude controller, then, a smooth position control is necessary to ensure the stability of the vehicle. A picture of the quad-rotor during a real-time experiment is depicted in Fig. 6.8. It can be seen that the UAV maintains its position relative to the landmark. A video of the quad-rotor while performing experiments can be seen at http://www.youtube.com/watch?v=SQlSXruTnj0.
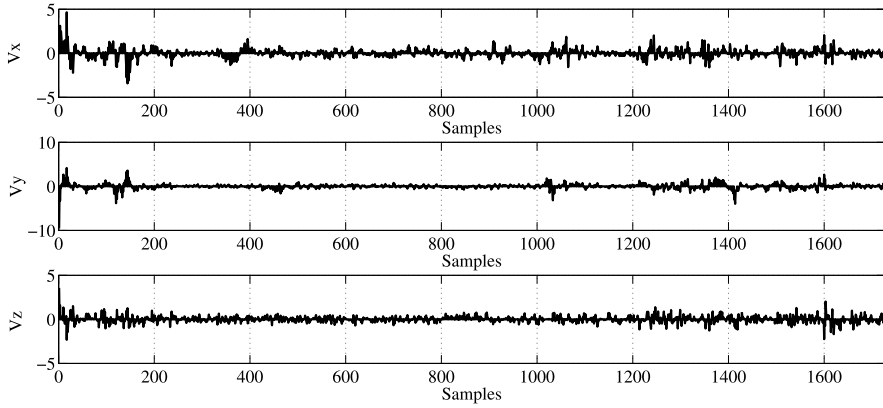
Fig. 6.6 Velocities, experimental results
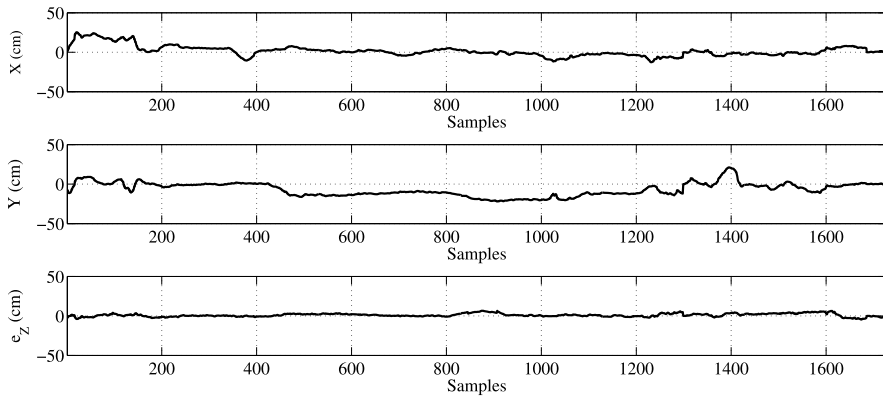
Fig. 6.7 Positions, experimental results

6.1.7 Final Comments

In this section, a vision-based strategy for stabilizing the 3-dimensional position of a quad-rotor with respect to a landing pad on the ground was proposed and tested in real-time experiments. The proposed vision algorithm consists of an homography estimation technique for extracting the 3-dimensional position, as well as of an optical flow computation for deriving the vehicle's translational velocities. A technique for predicting the landing pad position when its detection fails was presented also. The estimated information proved to be rich enough to allow performing autonomous missions. The experimental application was successfully performed indoors showing that the quad-rotor was stabilized at a selected (x, y, z) position above the landing pad. The attitude of the vehicle was not significantly perturbed by the vision-based control input used to correct the UAV position. The vehicle's velocities remained also very close to zero.

Fig. 6.8 The quad-rotor
stabilized over the landing
pad in a desired position

6.2 A Comparison of Nonlinear Controllers Using Visual Feedback

The previous section presented a vision-based stabilization of the quad-rotor using a hierarchical control strategy. Although the experimental results obtained demonstrate the effectiveness of such approach, it was considered that evaluating different control strategies in real-time experiments will help to determine the better control strategy for the research objectives. This reasoning motivated the present studies.

6.2.1 Introduction

Testing the performance of different controls over quad-rotors systems is a subject already studied. PID and LQR controllers are compared in [14], while in [15], the performance of a backstepping and a sliding modes controllers are tested. For the last two examples, experiments were performed over a quad-rotor platform, where three degrees of freedom are locked. The authors conclude that backstepping control technique is the most appropriate approach for their future work. In [6], two control methods are studied over a quad-rotor platform equipped with visual feedback. These methods are based on feedback linearization and a backstepping-like control.

The control objective of the studies presented in this Section consists of identifying the most effective controller for stabilizing the position of the "Cross-Flyer" UAV with respect to an artificial visual landmark placed on the ground. Three controllers considered between the most commonly reported in the literature have been chosen: nested saturations [27, 77], backstepping [70], and sliding modes [52]. The performance of such methodologies applied to the quad-rotor system is evaluated from real-time experimental results.

6.2.2 System Set-up

The vision-based position estimation approach used for these experiments is based on the same visual system set-up presented previously in Sect. 6.1.2. Similarly, visual feedback is provided from the imaging algorithms presented in Sect. 6.1.3, and the whole system composed by the "Cross-Flyer" UAV and supervisory ground station is the one presented in Sect. 6.1.5.

6.2.3 Control Strategies

The three control strategies applied to the quad-rotor stabilization are: nested saturations control method, backstepping approach, and sliding modes controller. The three of them are designed to stabilize the x, y, θ and ϕ states. The helicopter's altitude z and yaw angle ψ are stabilized by PD controllers, as shown in Sect. 6.1.4.1. The control algorithms were implemented onboard to stabilize the UAV's 3-dimensional position and attitude.

6.2.3.1 Nested Saturations Control

Consider a system given by four integrators in cascade:

$$\dot{x}_1 = \alpha x_2, \qquad \dot{x}_2 = \beta x_3, \qquad \dot{x}_3 = \gamma x_4, \qquad \dot{x}_4 = u \qquad (6.49)$$

where α, β, $\gamma \neq 0$ are constants. A nested saturations control input can be defined as [27, 77]:

$$u = -\sigma_{b_4}\left(k_4 z_4 + \sigma_{b_3}\left(k_3 z_3 + \sigma_{b_2}\left(k_2 z_2 + \sigma_{b_1}(k_1 z_1)\right)\right)\right) \qquad (6.50)$$

where z_i, for $i = 1, \ldots, 4$, denotes a change of variables. $k_i > 0$ are constants, and σ_{b_i} represent saturation functions:

$$\sigma_{b_i}(s) = \begin{cases} -b_i; & s < -b_i \\ s; & |s| \leq b_i \\ b_i; & s > b_i \end{cases} \qquad (6.51)$$

where $b_i > 0$ are constants denoting the bounds of the saturation functions. The z_i terms are given by

$$z_1 = x_4 + \frac{k_4 + k_3 + k_2}{\gamma} x_3 + \frac{k_3 k_4 + k_2 k_3 + k_2 k_4}{\beta \gamma} x_2 + \frac{k_2 k_3 k_4}{\alpha \beta \gamma} x_1 \qquad (6.52)$$

$$z_2 = x_4 + \frac{k_4 + k_3}{\gamma} x_3 + \frac{k_3 k_4}{\beta \gamma} x_2 \qquad (6.53)$$

$$z_3 = x_4 + \frac{k_4}{\gamma} x_3 \qquad (6.54)$$

$$z_4 = x_4 \qquad (6.55)$$

Note that from (6.11)–(6.12) and (6.15)–(6.16) $r_1 \to 0$ and $\psi \to \psi_d$. For a time T large enough, e_z and e_ψ are arbitrarily small, therefore, (6.13) and (6.14) reduce to

$$\ddot{x} = g \tan \theta \tag{6.56}$$

$$\ddot{y} = -g \frac{\tan \phi}{\cos \theta} \tag{6.57}$$

Control of Forward Position and Pitch Angle   Consider the subsystem given by (2.34) and (6.56). Implementing a nonlinear control based on nested saturations allows in the limit a guarantee of arbitrary bounds for x, $\dot{x}$, θ and $\dot{\theta}$. Let us impose a very small upper bound on $|\theta|$ in such a way that the difference $\tan(\theta) - \theta$ is arbitrarily small ($\theta \approx \tan(\theta)$). Therefore, the subsystem (2.34) and (6.56) becomes the following linearized system:

$$\ddot{x} = g\theta \tag{6.58}$$

$$\ddot{\theta} = \tilde{\tau}_\theta \tag{6.59}$$

which represents four integrators in cascade. Then, by using (6.49)–(6.50) the controller is

$$\tilde{\tau}_\theta = -\sigma_{b_4}\left(\dot{\theta} + \sigma_{b_3}\left(\dot{\theta} + \theta + \sigma_{b_2}\left(\dot{\theta} + 2\theta + \frac{\dot{x}}{g} + \sigma_{b_1}\left(\dot{\theta} + 3\theta + 3\frac{\dot{x}}{g} + \frac{x}{g}\right)\right)\right)\right) \tag{6.60}$$

This control law comes from the technique based on nested saturations control developed in [77]. It is proved in [27] that θ, $\dot{\theta}$, x and $\dot{x}$ converge to zero. To regulate x around a desired position, the most inner term (associated to σ_{b_1}) must be written as $\dot{\theta} + 3\theta + 3\frac{\dot{x}}{g} + \frac{e_x}{g}$, where e_x is the position error, expressed as $e_x = x - x_d$. Here, x_d represents the desired position reference for x.

Control of Lateral Position and Roll Angle   Consider the subsystem given by (2.35) and (6.57). Let us impose a very small upper bound on $|\phi|$ in such a way that the difference $\tan(\phi) - \phi$ is arbitrarily small ($\phi \approx \tan(\phi)$). Therefore, the subsystem (2.35) and (6.57) becomes

$$\ddot{y} = -g\phi \tag{6.61}$$

$$\ddot{\phi} = \tilde{\tau}_\phi \tag{6.62}$$

Using a similar procedure to the one proposed for the pitch control, we obtain

$$\tilde{\tau}_\phi = -\sigma_{b_4}\left(\dot{\phi} + \sigma_{b_3}\left(\dot{\phi} + \phi + \sigma_{b_2}\left(\dot{\phi} + 2\phi - \frac{\dot{y}}{g} + \sigma_{b_1}\left(\dot{\phi} + 3\phi - 3\frac{\dot{y}}{g} - \frac{y}{g}\right)\right)\right)\right) \tag{6.63}$$

In order to regulate y around a desired position, the most inner term (associated to σ_{b_1}) must be written as $\dot{\phi} + 3\phi - 3\frac{\dot{y}}{g} - \frac{e_y}{g}$, where e_y is the position error, expressed as $e_y = y - y_d$. Here, y_d represents the desired position reference for y.

6.2.3.2 Backstepping Control

The backstepping technique provides a systematic method to obtain a control law from a chain of integrators. This methodology was introduced in [70].

Control of Forward Position and Pitch Angle　　Rewrite subsystem given by (6.58)–(6.59) as

$$\dot{x}_1 = x_2 \tag{6.64}$$

$$\dot{x}_2 = g\theta_1 \tag{6.65}$$

$$\dot{\theta}_1 = \theta_2 \tag{6.66}$$

$$\dot{\theta}_2 = \tilde{\tau}_{B\theta} \tag{6.67}$$

where $\tilde{\tau}_{B\theta}$ will define the final backstepping control input. In order to obtain this control input, consider each equation as a new subsystem, where the next state is taken as the input and it is defined as a virtual control to stabilize such a subsystem. For the present case let us start with

$$\dot{x}_1 = x_2 \tag{6.68}$$

$$\zeta_1 = x_1 \tag{6.69}$$

where x_2 represents the input and ζ_1 the output. Let us propose a positive definite function $V_1 = \frac{1}{2}x_1^2$, whose time derivative is given by

$$\dot{V}_1 = x_1\dot{x}_1 = x_1 x_2 \tag{6.70}$$

and consider a virtual input $\alpha_1 = (x_2)^v = -k_1 x_1$, where k_1 is a positive constant. Then $\dot{V}_1 = -k_1 x_1^2$. Now, let ζ_2 be the new output:

$$\zeta_2 = x_2 - \alpha_1 \tag{6.71}$$

The new subsystem to be stabilized is written as

$$\dot{x}_1 = \zeta_2 + \alpha_1 \tag{6.72}$$

$$\dot{\zeta}_2 = g\theta_1 - \dot{\alpha}_1 \tag{6.73}$$

and let us propose a positive definite function $V_2 = V_1 + \frac{\zeta_2^2}{2}$, then

$$\dot{V}_2 = \dot{V}_1 + \zeta_2\dot{\zeta}_2 = \dot{V}_1 + \zeta_2(g\theta_1 - \dot{\alpha}_1) \tag{6.74}$$

Define a virtual input $\alpha_2 = (g\theta_1) = -k_2\zeta_2 + \dot{\alpha}_1$, where k_2 is a positive constant. Then, $\dot{V}_2 = \dot{V}_1 - k_2\zeta_2^2$. Now, let ζ_3 be a new output:

$$\zeta_3 = g\theta_1 - \alpha_2 \tag{6.75}$$

The new subsystem to be stabilized is written as

$$\dot{x}_1 = \zeta_2 + \alpha_1 \tag{6.76}$$

$$\dot{\zeta}_2 = g\theta_1 - \dot{\alpha}_1 \tag{6.77}$$

$$\dot{\zeta}_3 = g\theta_2 - \dot{\alpha}_2 \tag{6.78}$$

and let us propose the positive definite function $V_3 = V_2 + \frac{\zeta_3^2}{2}$, then

$$\dot{V}_3 = \dot{V}_2 + \zeta_3 \dot{\zeta}_3 = \dot{V}_2 + \zeta_3 (g\theta_2 - \dot{\alpha}_2) \qquad (6.79)$$

Define a virtual input $\alpha_3 = (g\theta_2) = -k_3\zeta_3 + \dot{\alpha}_2$, where k_3 is a positive constant. Then, $\dot{V}_3 = \dot{V}_2 - k_3\zeta_3^2$. Let ζ_4 be the new output:

$$\zeta_4 = g\theta_2 - \alpha_3 \qquad (6.80)$$

The new subsystem to be stabilized is written as

$$\dot{x}_1 = \zeta_2 + \alpha_1 \qquad (6.81)$$
$$\dot{\zeta}_2 = g\theta_1 - \dot{\alpha}_1 \qquad (6.82)$$
$$\dot{\zeta}_3 = g\theta_2 - \dot{\alpha}_2 \qquad (6.83)$$
$$\dot{\zeta}_4 = g\dot{\theta}_2 - \dot{\alpha}_3 \qquad (6.84)$$

and let us propose the Lyapunov candidate function $V_4 = V_3 + \frac{\zeta_4^2}{2}$, then

$$\dot{V}_4 = \dot{V}_3 + \zeta_4 \dot{\zeta}_4 = \dot{V}_3 + \zeta_4 (g\tilde{\tau}_{B\theta} - \dot{\alpha}_3) \qquad (6.85)$$

Let us propose the backstepping control input $\tilde{\tau}_{B\theta}$ as

$$\tilde{\tau}_{B\theta} = \frac{1}{g}(-k_4\zeta_4 + \dot{\alpha}_3) \qquad (6.86)$$

where k_4 is a positive constant. Then

$$\dot{V}_4 = \dot{V}_3 - k_4\zeta_4^2 = -k_1 x_1^2 - k_2\zeta_2^2 - k_3\zeta_3^2 - k_4\zeta_4^2 \qquad (6.87)$$

With the proposed backstepping control input $\tilde{\tau}_{B\theta}$, one has $\dot{V}_4 < 0$, and then the system (6.64)–(6.67) is globally asymptotically stable. In order to express $\tilde{\tau}_{B\theta}$ as a function of x_1, x_2, θ_1 and θ_2, ζ_4 and $\dot{\alpha}_3$ must be rewritten as a function of such variables:

$$\zeta_4 = g\theta_2 + (k_1 + k_2 + k_3)g\theta_1 + (k_1 k_2 + k_1 k_3 + k_2 k_3)x_2 + k_1 k_2 k_3 x_1 \quad (6.88)$$
$$\dot{\alpha}_3 = -(k_1 + k_2 + k_3)g\theta_2 - (k_1 k_2 + k_1 k_3 + k_2 k_3)g\theta_1 - k_1 k_2 k_3 x_2 \quad (6.89)$$

The final control input is rewritten as

$$\tilde{\tau}_{B\theta} = -\frac{\overline{k}_1}{g}x_1 - \frac{\overline{k}_2}{g}x_2 - \overline{k}_3\theta_1 - \overline{k}_4\theta_2 \qquad (6.90)$$

where

$$\overline{k}_1 = k_1 k_2 k_3 k_4 \qquad (6.91)$$
$$\overline{k}_2 = k_1 k_2 k_3 + k_1 k_2 k_4 + k_1 k_3 k_4 + k_2 k_3 k_4 \qquad (6.92)$$
$$\overline{k}_3 = k_1 k_2 + k_1 k_3 + k_1 k_4 + k_2 k_3 + k_2 k_4 + k_3 k_4 \qquad (6.93)$$
$$\overline{k}_4 = k_1 + k_2 + k_3 + k_4 \qquad (6.94)$$

Control of Lateral Position and Roll Angle   Rewrite the subsystem given by (6.61)–(6.62) as $\dot{y}_1 = y_2$, $\dot{y}_2 = -g\phi_1$, $\dot{\phi}_1 = \phi_2$ and $\dot{\phi}_2 = \tilde{\tau}_{B\phi}$. Using a similar procedure to the one proposed for the pitch control, the backstepping roll control input can be obtained as

$$\tilde{\tau}_{B\phi} = \frac{\overline{k}_1}{g}y_1 + \frac{\overline{k}_2}{g}y_2 - \overline{k}_3\phi_1 - \overline{k}_4\phi_2 \tag{6.95}$$

6.2.3.3 Sliding Modes Control

Consider the system (6.49) of integrators in cascade, and rewrite this system in the form [52]

$$\begin{bmatrix} \dot{\boldsymbol{\varphi}} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_a(\varphi, \varepsilon) \\ f_b(\varphi, \varepsilon) + G(x)E(x)u + \delta(t, x, u) \end{bmatrix} \tag{6.96}$$

then, $\boldsymbol{\varphi} = [x_1 \; x_2 \; x_3]^T$ and $\varepsilon = x_4$. Also $\mathbf{f}_a(\boldsymbol{\varphi}, \varepsilon) = [\alpha x_2 \; \beta x_3 \; \gamma x_4]^T$ and $f_b(\boldsymbol{\varphi}, \varepsilon) = 0$, $G(x) = E(x) = 1$, $\delta(t, x, u) = 0$. Consider $x_4 = -c_0 x_1 - c_1 x_2 - c_2 x_3 = \phi(\boldsymbol{\varphi})$ in order to stabilize the origin. The partial derivative of ϕ, with respect to $\boldsymbol{\varphi}$, is given by

$$\frac{\partial \phi}{\partial \boldsymbol{\varphi}} = [-c_0 \quad -c_1 \quad -c_2] \tag{6.97}$$

then

$$\frac{\partial \phi}{\partial \boldsymbol{\varphi}}\mathbf{f}_a = -c_0\alpha x_2 - c_1\beta x_3 - c_2\gamma x_4 \tag{6.98}$$

Define a sliding surface

$$s = \varepsilon - \phi(\boldsymbol{\varphi}) = x_4 + c_0 x_1 + c_1 x_2 + c_2 x_3 = 0 \tag{6.99}$$

whose time derivative is given by

$$\dot{s} = -\frac{\partial \phi}{\partial \boldsymbol{\varphi}}\mathbf{f}_a(\boldsymbol{\varphi}, \varepsilon) + u \tag{6.100}$$

then, the control input can be expressed as

$$u = -c_0\alpha x_2 - c_1\beta x_3 - c_2\gamma x_4 + v \tag{6.101}$$

where $\dot{s} = v$ and $v = -\nu(x)\tanh(\frac{s}{\varepsilon})$. By definition, one has

$$\dot{s} = g(x)v + \Delta(t, x, u) \tag{6.102}$$

thus $g(x) = 1$ and $\Delta(t, x, u) = 0$. Also, one needs to satisfy

$$\left| \frac{\Delta(t, x, u)}{g(x)} \right| \leq \rho(x) + k_0\|v\| \tag{6.103}$$

then

$$\left| \frac{0}{1} \right| \leq B \tag{6.104}$$

where B is a positive constant, thus $\rho(x) = B$ and $k_0 = 0$. One has then

$$v(x) \geq \frac{\rho(x)}{1 - k_0} \geq B \tag{6.105}$$

Now it can be determined that $v = -v(x)\tanh(\frac{s}{\varepsilon})$ or $v = -B\tanh(\frac{s}{\varepsilon})$. Note that $\tanh(s/\varepsilon)$ is a smooth approximation of the function $\mathrm{sign}(s)$, which is used in order to reduce the chattering effect. ε is selected as a small constant. Finally, it is now possible to completely write the input signal equation as

$$u = -c_0\alpha x_2 - c_1\beta x_3 - c_2\gamma x_4 - B\tanh\left(\frac{s}{\varepsilon}\right) \tag{6.106}$$

For the forward-pitch subsystem equations given by (6.58)–(6.59), and by analogy with the system (6.49), one obtains $\alpha = 1$, $\beta = g$, and $\gamma = 1$, also, one has that $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$. With this information, the sliding mode surface is given by

$$s = c_0 x + c_1 g\dot{x} + c_2\theta + \dot{\theta} \tag{6.107}$$

and the forward-pitch control input can be expressed as

$$\tilde{\tau}_\theta = -c_0\dot{x} - c_1\theta - c_2\dot{\theta} - B\tanh\left(\frac{s}{\varepsilon}\right) \tag{6.108}$$

The constant terms c_i should be carefully selected to obtain a stable output. In order to stabilize x in a position outside of the origin, one must place $x_1 = e_x$.

Following a similar procedure the lateral-roll control input is given by

$$\tilde{\tau}_\phi = -c_0\dot{y} - c_1\phi - c_2\dot{\phi} - B\tanh\left(\frac{s}{\varepsilon}\right)$$

6.2.4 Experimental Applications

During this studies, three similar experiments were performed, with the purpose of identifying the most appropriate control strategy for stabilizing the position of the quad-rotor in hover flight. The experiments followed a similar procedure than the one described in Sect. 6.1.6.

Once the UAV is located exactly on top of the landing pad, the operator uses the joystick of the supervisory ground station to define the current vehicle's position as the desired (x_d, y_d) position reference. The desired altitude z_d is always fixed at 150 cm, and the desired yaw angle ψ is fixed at $0°$ degrees. The parameter values used for the altitude and the yaw controller are: $k_{pz} = 0.68$, $k_{vz} = 1.6$, $k_{p\psi} = 38$, $k_{p\psi} = 1350$. The parameters used for the nested saturations controller are: $\sigma_{b_4} = 0.4700$, $\sigma_{b_3} = 0.2349$, $\sigma_{b_2} = 0.1174$, and $\sigma_{b_1} = 0.0287$. The parameters used for
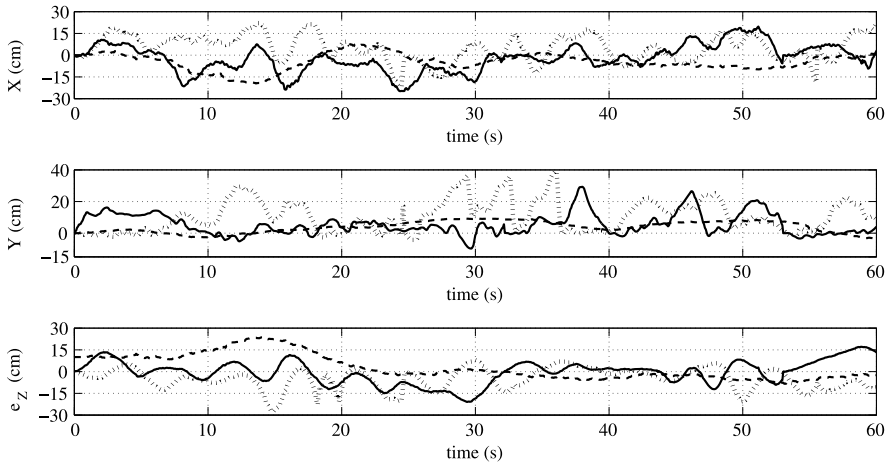
Fig. 6.9 Behavior of the x, y, and altitude error signals. The backstepping, nested saturation and sliding modes are represented by *solid line*, *dashed line*, and *dotted line*, respectively

the backstepping controller are: $\overline{k}_1 = 0.002$, $\overline{k}_2 = 0.004$, $\overline{k}_3 = 9$, and $\overline{k}_4 = 4$. The parameters used for sliding mode controller are: $c_0 = 0.3$, $c_1 = 0.25$, $c_2 = 0.15$, $B = 0.011$, and $\varepsilon = 0.05$. Those control parameters were found by trial and error.

Figures 6.9, 6.10, 6.11 and 6.12 show the obtained behavior when applying the three controllers to the UAV. It can be observed that all the controllers achieve hovering flight, however, smoother translational and angular behavior is obtained when using the nested saturations controller. One of the advantages of the nested saturation control technique is that it has a smooth behavior. Indeed, the saturation functions are not introducing jumps in the control input, and after a finite time the system will operate as a linear system. Furthermore, the nested saturation technique allows dealing first with the angular dynamics, which is the most important part of the vehicle stabilization, and once this is done we can deal with the stabilization of the translational dynamics. Figures 6.9, 6.10, 6.11 and 6.12, show that the backstepping and the sliding modes controllers induce faster changes in the vehicle's attitude, which as a consequence degrade the 3-dimensional position stabilization of the quad-rotor during the real-time experiments.

Tables 6.1 and 6.2 show the mean and standard deviation values for the position and Euler angles signals, respectively. Note that Tables 6.1 and 6.2 were computed with only one experiment for each controller, considering that the UAV is in steady state response. Mean square errors for the Euler angles were also computed, the values are shown in Table 6.3. These results show that the nested saturations controller is the method that induces less angular corrections, which can be considered as less control input generated during flight, and consequently, less energy consumption. Note in Tables 6.1 and 6.2 that the values for the backstepping controller are closer to the desired reference values. However, if the important objective concerns the energy consumption, the nested saturations controller should be considered as the best option.
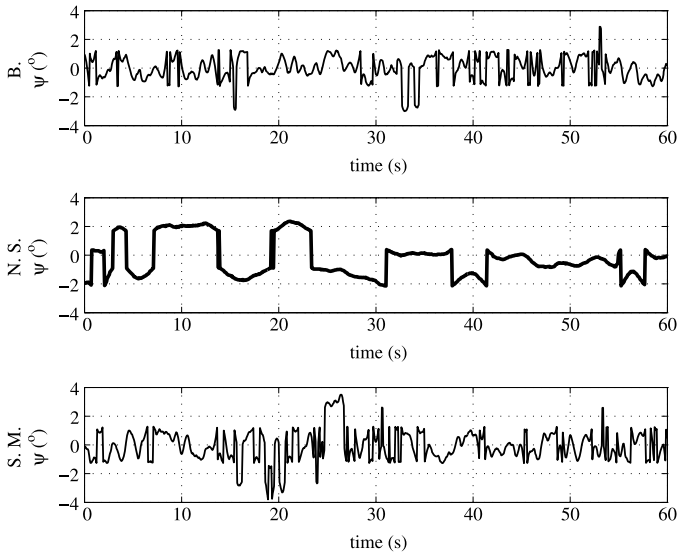
Fig. 6.10 Behavior of the yaw angle
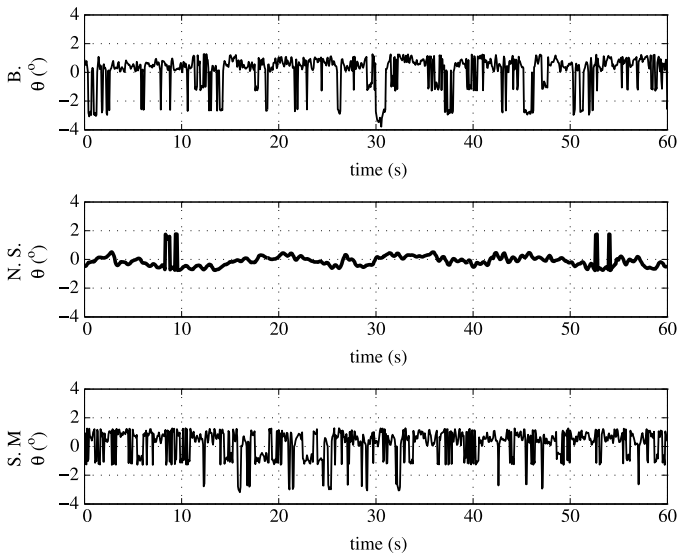
Fig. 6.11 Behavior of the pitch angle

6.2.5 Final Comments

From the experiments conducted during these studies, it has been observed that the
three control algorithms ensure that the Euler angles of the vehicle remain very
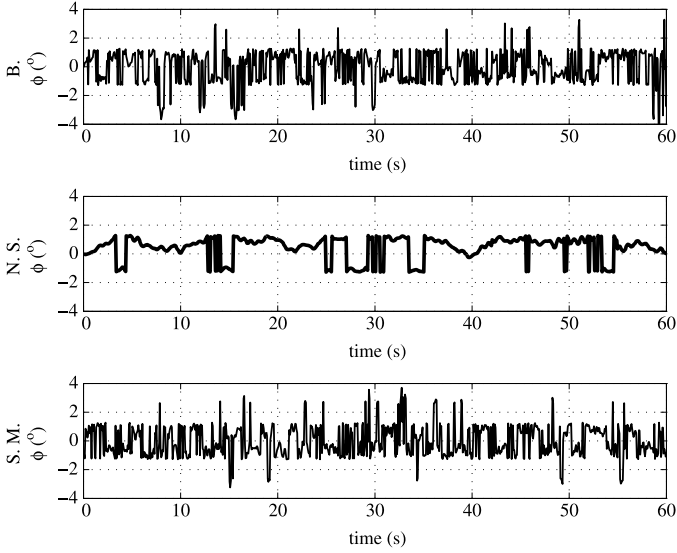
Fig. 6.12 Behavior of the roll angle

Table 6.1 Mean values of position and Euler angles

Parameter	Backstepping	Nested saturations	Sliding modes
Yaw angle	−0.0108°	−0.2504°	−0.0400°
Pitch angle	0.0903°	−0.0706°	0.2033°
Roll angle	−0.0315°	0.4371°	−0.0782°
e_Z error	0.0176 cm	1.4550 cm	−6.8905 cm
X position	−1.5662 cm	−2.7342 cm	4.3211 cm
Y position	5.6848 cm	4.2451 cm	10.1918 cm

Table 6.2 Standard deviation of position and Euler angles

Parameter	Backstepping	Nested saturations	Sliding modes
Yaw angle	0.7968°	1.2606°	1.0260°
Pitch angle	1.1588°	0.4473°	0.9833°
Roll angle	1.1673°	0.7635°	1.0739°
e_Z error	8.0868 cm	8.2162 cm	7.7071 cm
X position	9.2075 cm	6.2887 cm	10.2763 cm
Y position	7.1207 cm	4.1411 cm	9.2603 cm

close to the desired values. The obtained results show that the nested saturations control approach is the most appropriated strategy for our system, since it ensures a smoother vehicle behavior and reduces the energy consumption with respect to the other two controllers.

Table 6.3 Mean square error for Euler angles

Parameter	Backstepping	Nested saturations	Sliding modes
Yaw angle	0.6350°	1.5910°	1.0853°
Pitch	1.3519°	0.1476°	1.0144°
Roll	1.3499°	0.7623°	1.1343°

6.3 Vision-Based Altitude and Velocity Regulation

This section deals with achieving a hover flight and velocity regulation of a quad-rotor with the purpose of performing autonomous navigation. A vision system has been designed, which estimates the altitude, the lateral position and the forward speed of the helicopter during flights. It is shown that the visual information allows the construction of control strategies for different kinds of flying mode, for example hover flight and forward flight at constant speed. Experimental autonomous flights validate the visual algorithm and the control law.

6.3.1 Introduction

In this section, a monocular vision system is implemented onboard a UAV allowing altitude control, with the objective of stabilizing the position and regulating the velocity of the vehicle using optical flow. The objective of this study is to eliminate the position drift when hovering, as well as ensuring accurate displacements, two fundamental characteristics for any autonomous navigation system. If the position drift is compensated, the hover flight could be used by the system as an intermediary task between different flying modes, adapted to different conditions of the environment. Moreover, speed regulation is implemented to create the different flying modes, for example, only lateral displacement and only forward displacement. To correctly use the optical flow, a vision-based altitude controller has been also developed. The combination of these three vision-based controls allows the vehicle to navigate in a realistic application. The system is tested in two different kinds of mission: position hold over a road segment and road-following.

6.3.2 System Set-up

The overall system consists of the "Cross-Flyer" UAV platform with an embedded camera pointing downwards and the road model, as shown in Fig. 6.13. The relationship between the different reference frames can be summarized as follows:

- **Quad-rotor UAV:** evolving in space, with 6DOF. It is related to its own body coordinate system (X_B, Y_B, Z_B) and to the fixed inertial frame (X_I, Y_I, Z_I). The
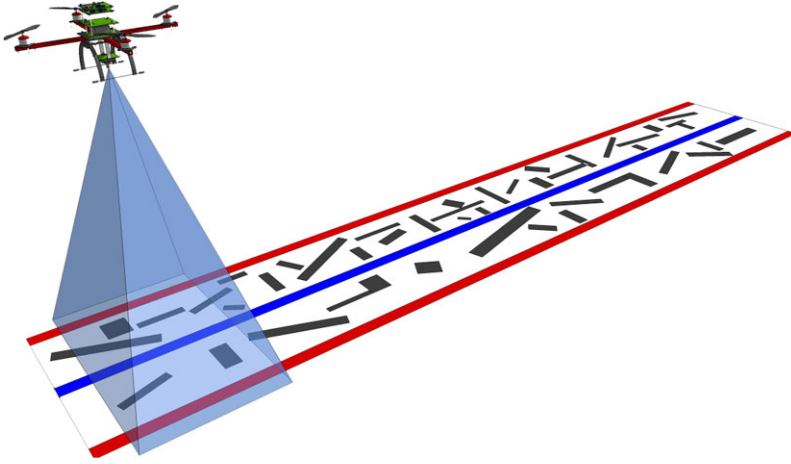
Fig. 6.13 Navigation: Visual system set-up

inertial coordinate system is located at the left corner of the beginning of the road, with the axis Z_I pointing upwards, the Y_I axis parallel to the road and the axis X_I perpendicular and coplanar with Y_I. The body coordinate system is set in the center of the rotorcraft, with Z_B the yaw axis, X_B the pitch axis and Y_B the roll axis. The planes formed by (X_B, Y_B) and (X_I, Y_I) are considered parallel since the attitude stabilization keeps the pitch and roll angles near zero.

- **Pointing downwards camera:** Attached to the rotorcraft, the camera undergoes the same movement. Any sensed motion represents a displacement of the helicopter. The center of the coordinate system is located at the center of the camera, and the axis Z_C is in the opposite direction with respect to Z_I and Z_B.
- **Road model:** Represented in Figs. 6.13 and 6.14 is a dual carriageway road, with a crash barrier separating both carriageway. The crash barrier is modeled as a blue line of width K_b. Each border of the road is modeled by a red line of width K_r. The width of the road is constant and equal to W.

The imaging algorithm computes an estimate of altitude, the vehicle's position and velocities. The altitude and some position information of the engine are deduced by extracting the road zone from the image, the velocities are estimated using optical flow. All information sensed by the camera is related to the image plane, it will be shown that this information is rich enough to control the vehicle.

6.3.3 Image Processing

For a correct utilization of the optical flow, fly at a constant altitude is fundamental. To guarantee this condition, the proposed imaging algorithm computes an estimation of the altitude. Once the altitude is estimated, the vision-based velocity control can be implemented.

Fig. 6.14 The road model
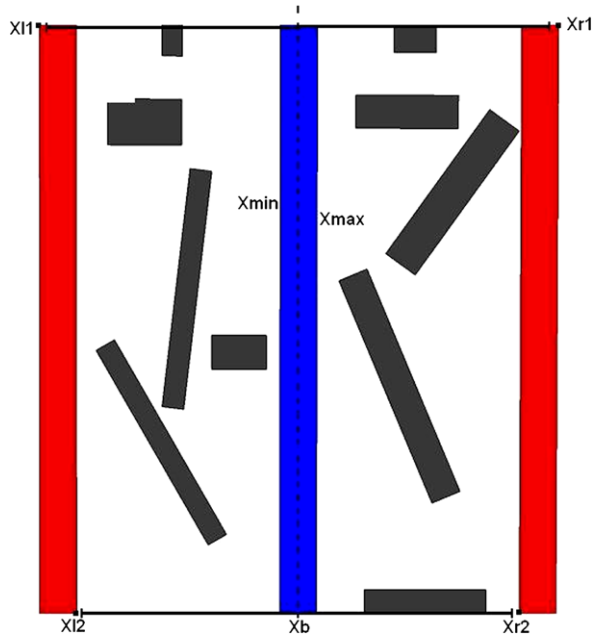
6.3.3.1 Extracting the Road Zone

The first step to deduce the helicopter altitude consists of detecting the road zone, which is achieved by using a combination of line detection and color segmentation processes. The first step is to recognize the color zones by means of a classical color segmentation algorithm performed in the RGB encoding format. A threshold level τ_r between the three colors defines the red color, and a threshold τ_b defines the blue color. Once the segmentation task is done, the image is passed through a series of Gaussian filters, then, the algorithm continues with a line detection process based on a Hough transform. Finally, both results are combined to estimate the points $x^i_{l_1}$, $x^i_{l_2}$, $x^i_{\min}$, $x^i_{\max}$, $x^i_{r_1}$, and $x^i_{r_2}$. These points are image projections of the world coordinates x_{l_1}, x_{l_2}, $x_{\min}$, $x_{\max}$, x_{r_1}, and x_{r_2} representing the left border, the central separation and the right border of the road, see Fig. 6.14. With all extracted image points, the road zone is totally defined and the altitude and the position can be estimated.

6.3.3.2 Altitude and Position

The altitude estimation is done using both lines passing through the points $x^i_{l_1}$, $x^i_{l_2}$, $x^i_{r_1}$, $x^i_{r_2}$. The importance of using these points lies on having an estimate, independent of the yaw angle of the vehicle, with respect to the inertial frame (coordinates system of the road). Let us define the following variables:

$$x_{l_m}^i = \frac{x_{l_2}^i + x_{l_1}^i}{2} \tag{6.109}$$

$$x_{r_m}^i = \frac{x_{r_2}^i + x_{r_1}^i}{2} \tag{6.110}$$

$$\omega = x_{r_m}^i - x_{l_m}^i \tag{6.111}$$

where $x_{l_m}^i$, $x_{r_m}^i$ represent the border's centroids of the left and right carriageways of
the road, respectively. Those points are in the image plane, and are related to the
world points by the projective camera relations. Considering that all points have the
same depth z, one can write

$$x_{l_m}^i = \frac{f\,x_l}{z} \tag{6.112}$$

$$x_{r_m}^i = \frac{f\,x_r}{z} \tag{6.113}$$

Introducing (6.112) and (6.113) in (6.111), one has

$$\omega = \frac{f(x_r - x_l)}{z} = \frac{f\,W}{z} \tag{6.114}$$

where W is the distance between the left and right boarders, i.e. the width of the
road. Then

$$\frac{1}{\omega} = K_{im}z \tag{6.115}$$

where K_{im} is a constant depending on the focal length of the camera and the width
of the road, and z is the altitude of the engine. From (6.115), the control of the
inverse of the image-based altitude variable ω is equivalent to control the altitude z
of the helicopter. The lateral position of the helicopter can also be deduced from the
road extraction points. The lines passing through the points $x_{\min}^i$, $x_{\max}^i$ defines the
new centroid x^i, which is the projection of the point x_b, see Fig. 6.14. The objective
is to bring the image point x^i to the center of the image, which is close enough to
the principal point of the image. This point can be considered as the projection of
the vehicle's center of gravity, since the center of the camera and the center of the
helicopter's coordinate system are aligned. Therefore, the new image-based variable
can be constructed as follows:

$$\varsigma = \frac{f(x - x_b)}{z} \tag{6.116}$$

where x is the position of the helicopter's center of gravity in the inertial frame,
x_b is the central line centroid, and ς is the image-based lateral position variable.
A study of the time derivative is needed in order to measure the impact of an altitude
variation on this variable,

$$\dot{\varsigma} = \frac{f}{z}\dot{x} - \frac{f(x - x_b)}{z^2}\dot{z} \tag{6.117}$$

$$\dot{\varsigma} = \frac{f}{z}\dot{x} + \frac{(x - x_b)}{W}\dot{\omega} \tag{6.118}$$

where $\dot{x}$ is the lateral speed of the helicopter, $\dot{\varsigma}$ is the time derivative of the image-based lateral position variable, and $\dot{\omega}$ is the time derivative of the image-based altitude variable. From (6.118) it is deduced that a previous stabilization of $\dot{\omega}$ is a necessary requirement to fulfill a good lateral position stabilization. Once the altitude is maintained constant, the time derivative $\dot{\varsigma}$ will only depend of the lateral speed of the helicopter, any other contribution will be considered as noise. Finally, when the altitude is stabilized $\dot{\omega} = 0$, then the time derivative of the image-based variables becomes

$$\dot{\varsigma} = \frac{f}{z}\dot{x} \tag{6.119}$$

6.3.3.3 Translational Velocities

Using a similar procedure as the one followed in Sect. 6.1.3.2, let us consider the camera moving with respect to a rigid scene. The velocities and rotation rates of the camera in the inertial frame are expressed by (V_x, V_y, V_z) and (w_x, w_y, w_z), respectively. To accurately estimate the pseudo-velocities of the engine, a tracking zone inside the road model is defined in a way that the point $x^i = f\frac{xb}{z}$ is the centroid of the zone. The optical flow computed at point (x_k^i, y_k^i) is composed of a translational and a rotational part as

$$\begin{bmatrix} \mathrm{OF}_{x_k^i} \\ \mathrm{OF}_{y_k^i} \end{bmatrix} = \mathbf{T}_{\mathrm{OF}_k} + \mathbf{R}_{\mathrm{OF}_k} \tag{6.120}$$

with the translational part

$$\mathbf{T}_{\mathrm{OF}_k} = \frac{1}{z}\begin{bmatrix} -f & 0 & x_k^i \\ 0 & -f & y_k^i \end{bmatrix}\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{6.121}$$

and the rotational part

$$\mathbf{R}_{\mathrm{OF}} = \begin{bmatrix} \frac{x_k^i y_k^i}{f} & -(f + \frac{(x_k^i)^2}{f}) & y_k^i \\ (f + \frac{(y_k^i)^2}{f}) & -\frac{x_k^i y_k^i}{f} & -x_k^i \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6.122}$$

The spatial mean of the optical flow (mean of the optical flow on all tracked points) is then

$$\bar{\mathrm{OF}}_x = \bar{V}_{\mathrm{OF}_x} + K_x^x \bar{V}_{\mathrm{OF}_z} + \bar{R}_{\mathrm{OF}_x} \tag{6.123}$$

$$\bar{\mathrm{OF}}_y = \bar{V}_{\mathrm{OF}_y} + K_y^y \bar{V}_{\mathrm{OF}_z} + \bar{R}_{\mathrm{OF}_y} \tag{6.124}$$

Using the results presented in [68], the rotational optical flow is compensated and the pseudo-speeds $(\bar{V}_{\mathrm{OF}_x}, \bar{V}_{\mathrm{OF}_y}, \bar{V}_{\mathrm{OF}_z})$ are deduced. Since the camera system and the helicopter share the same movements, one can write

$$\bar{V}_{\text{OF}_x} = -\frac{f\dot{x}}{z} = -\dot{\varsigma} \tag{6.125}$$

$$\bar{V}_{\text{OF}_y} = -\frac{f\dot{y}}{z} = -\dot{y}^i \tag{6.126}$$

$$\bar{V}_{\text{OF}_z} = \frac{\dot{z}}{z} \tag{6.127}$$

where $(\dot{x}, \dot{y}, \dot{z})$ is the speed vector of the rotorcraft center of gravity. From (6.115), (6.125), and (6.126), note that the visual system allows altitude, position and velocity control of the rotorcraft using only the image-based variables $(\varsigma, \dot{\varsigma})$, $(\frac{1}{\omega}, (\frac{\dot{1}}{\omega}))$ and $\dot{y}^i = -\bar{V}_{\text{OF}_y}$.

6.3.4 Control Strategy

In this study, a similar hierarchical control strategy than the one presented in Sect. 3.5 was implemented. The first two subsystems to be stabilized are the altitude and the yaw dynamics.

6.3.4.1 Altitude and Yaw Subsystems

The vision-based altitude controller is deduced from (6.11) and (6.115), then, the altitude can be stabilized with the feedback input

$$u = \frac{m(g - k_1^{\omega}(\frac{1}{\omega} - \frac{1}{\omega_{\text{ref}}}) - k_2^{\omega}\frac{d}{dt}\frac{1}{\omega})}{\cos\theta\cos\phi} \tag{6.128}$$

where $\frac{1}{\omega}$ is the visual estimation of the altitude, and $\frac{d}{dt}\frac{1}{\omega}$ is computed by numerical differentiation of the variable $\frac{1}{\omega}$. For the design of this control law, the pseudo-speed in the direction of the optical axis $\bar{V}_{\text{OF}_z}^d$, estimated with the optical flow algorithm, is not used. When u is implemented, the dynamic of the altitude becomes

$$\ddot{z} = \cos\theta\cos\phi\frac{u}{m} - g \tag{6.129}$$

$$\ddot{z} = -k_1^{\omega}\left(\frac{1}{\omega} - \frac{1}{\omega_{\text{ref}}}\right) - k_2^{\omega}\frac{d}{dt}\frac{1}{\omega} \tag{6.130}$$

Then, by choosing the gains $k_1^{\omega} = K_{im}k_p^z$ and $k_2^{\omega} = K_{im}k_d^z$, the closed-loop dynamics can be seen as follows:

$$\ddot{z} = -k_p^z(z - z_{\text{ref}}) - k_d^z\dot{z} \tag{6.131}$$

The gains k_p^z and k_d^z are chosen to ensure that the polynomial $s^2 + k_d^z s + k_p^z$ is Hurwitz. If this assumption is validated, the altitude will converge to the reference value z_{ref}. Once the altitude converges, the longitudinal and the lateral subsystems

can be seen as linear subsystems. For the yaw dynamics, one has an independent system composed of two integrators in cascade. In order to ensure the stabilization of this subsystem, the following control strategy is proposed:

$$\tilde{\tau}_\psi = -k_2^\psi \dot{\psi} - k_1^\psi \psi \tag{6.132}$$

The gains k_1^ψ and k_2^ψ are chosen appropriately so that the polynomial $s^2 + k_2^\psi + k_1^\psi$ is Hurwitz and the yaw dynamics converges to zero.

6.3.4.2 Longitudinal Subsystem

The closed-loop dynamics of θ, viewed by the navigation control, becomes

$$\dot{y}_1 = y_2^{\text{ref}} + \tilde{y}_2 \tag{6.133}$$
$$\dot{\tilde{y}}_2 = \tilde{y}_3 \tag{6.134}$$
$$\dot{\tilde{y}}_3 = \tilde{\tau}_\theta \tag{6.135}$$

one chooses

$$y_2^{\text{ref}} = -k_1\left(\bar{V}_{\text{OF}_y}^d - \bar{V}_{\text{ref}}^d\right) \tag{6.136}$$

which leads to

$$y_2^{\text{ref}} = -k_p^y y_1 \tag{6.137}$$

and

$$\tilde{\tau}_\theta = -k_d^\theta \dot{\theta} - k_p^\theta (\theta - \theta_{\text{ref}}) \tag{6.138}$$

where $\theta_{\text{ref}} = -k_1(\bar{V}_{\text{OF}_y}^d - \bar{V}_{\text{ref}}^d)$. The closed-loop system is represented by the polynomial $s^3 + k_d^\theta s^2 + k_p^\theta s + k_p^\theta k_p^y$, which is Hurwitz if $k_d^\theta, k_p^\theta, k_p^y > 0$ and $k_d^\theta > k_p^y$. Then, the subsystem $\dot{y} - \theta$ converges to the desired references.

6.3.4.3 Lateral Subsystem

The implementation of the vision-based control to the lateral dynamics gives

$$\dot{x}_1 = x_2 \tag{6.139}$$
$$\dot{x}_2 = x_3^{\text{ref}} + \tilde{x}_3 \tag{6.140}$$
$$\dot{\tilde{x}}_3 = x_4 \tag{6.141}$$
$$\dot{x}_4 = \tilde{\tau}_\phi \tag{6.142}$$

where $x_3^{\text{ref}} = \phi_{\text{ref}}$. This reference angle is computed by using $(\varsigma, \dot{\varsigma})$ from (6.117) and (6.123), respectively. The reference angle dynamics are ignored since the rotational dynamics of the vehicle are faster than the translational dynamics and also because

the low-level autopilot was designed using high gains. The image-based variables $(\varsigma, \dot{\varsigma})$ are used to stabilize the position, as follows:

$$x_3^{\text{ref}} = \phi_{\text{ref}} = -k_2^x \dot{\varsigma} - k_1^x \varsigma \tag{6.143}$$

$$\phi_{\text{ref}} = -k_d^x \dot{x} - k_p^x (x - x_b) \tag{6.144}$$

Then

$$\tilde{\tau}_\phi = -k_d^\phi \dot{\phi} - k_p^\phi (\phi - \phi_{\text{ref}}) \tag{6.145}$$

The closed-loop system is stable if $k_p^\phi, k_d^\phi, k_p^x, k_d^x > 0$, $k_d^\phi > k_d^x$, and $k_p^\phi > \frac{(k_d^\phi)^2 k_p^x}{k_d^x (k_d^\phi - k_d^x)}$. Finally, the roll and the lateral displacement converge to their references.

6.3.5 Experimental Application

The experimental implementation is realized in the "Cross-Flyer" UAV platform, with the visual system detailed in Sect. 5.4.3. Two main experiments were performed: hover flight and forward flight at constant speed. The goal of such experiments is to validate the vision-based control strategy described proposed in this section.

The first experiment consists on a hover flight stabilization, over a specified zone of the road model. The hover flight is activated after a manual take-off. During experiments, the total width of the road model needs to be under the field of view of the camera in order to estimate the variable $\frac{1}{\omega}$. For this reason, the control strategy is activated once the vehicle is already flying at a desired altitude, stabilizing the rotorcraft at the current position. Figures 6.15, 6.16, and 6.17 show the behavior of the vehicle's states during real-time experiments. It can be observed in Fig. 6.17 that the Euler angle measured remains small. A video of the quad-rotor while performing such experiment can be seen at http://www.youtube.com/watch?v=xPb-IHSsNIo& feature=related.

For the second experiment, the vehicle is first set on manual mode and positioned exactly over the road model, then, the automatic speed regulation is activated via the joystick control mode. This allows the vision-based system to take the control of the quad-rotor. During tests, the vehicle sends all the information needed to analyze the experiment to the base station. The set of Figs. 6.18, 6.19, and 6.20 show the behavior of the helicopter position, velocities and Euler angles, respectively, for the velocity regulation experiment. Figure 6.19 shows how the lateral velocity is kept near zero, while the forward velocity converges to the desired value, which was set to a value of 5. This value is not the real velocity but the optical flow value in the forward direction. The altitude of the engine is also satisfactorily stabilized. It can be seen in Figs. 6.18 and 6.19 that the altitude z and velocity $\dot{z}$ present only small changes. A video of the UAV flying forward at constant speed can be seen at http://www.youtube.com/watch?v=PpUW9a3S3GQ&feature=related.
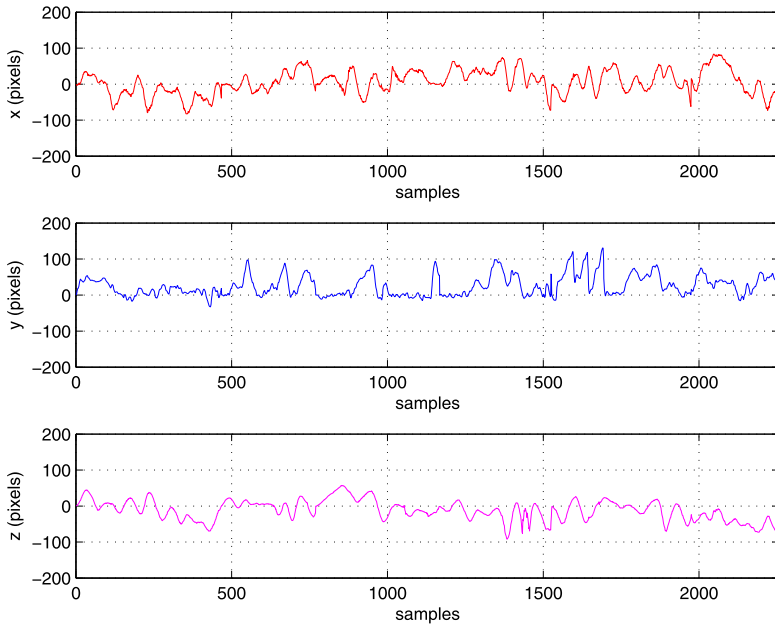
Fig. 6.15   Hover stabilization: Position results
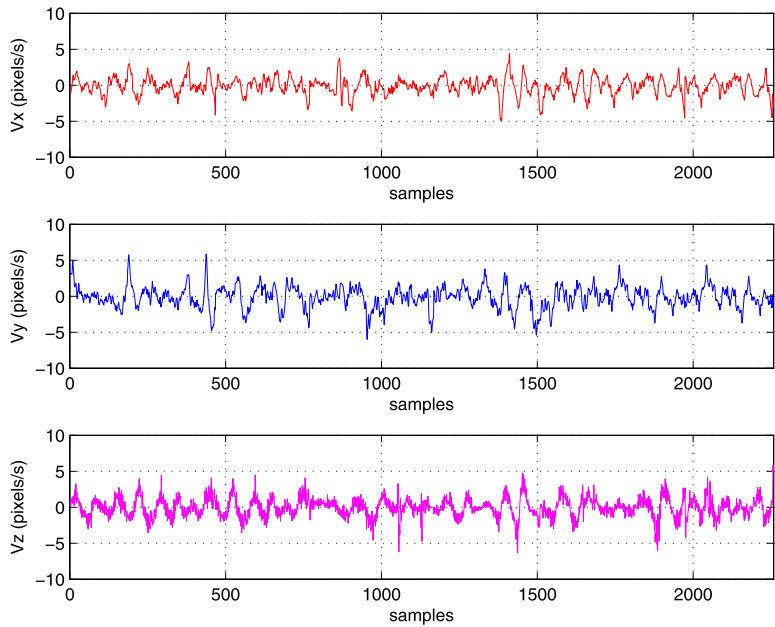
Fig. 6.16   Hover stabilization: Velocities results
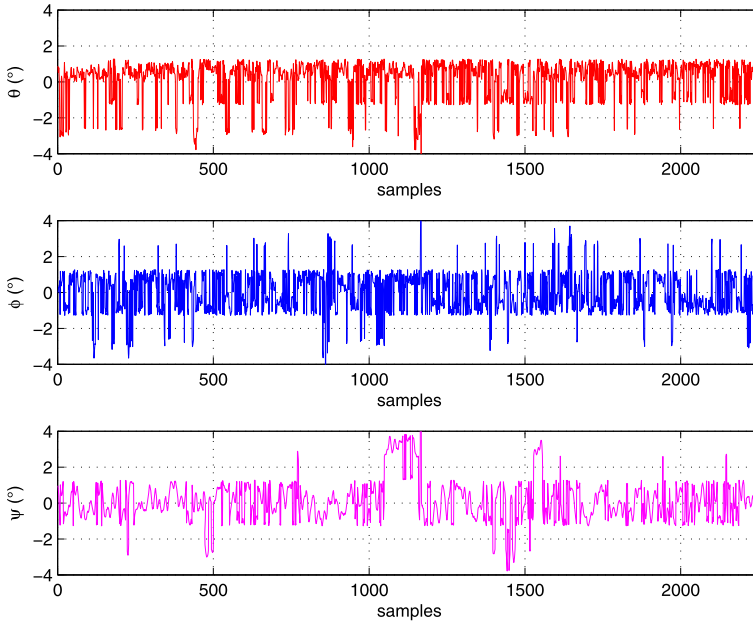
Fig. 6.17 Hover stabilization: Euler angles results
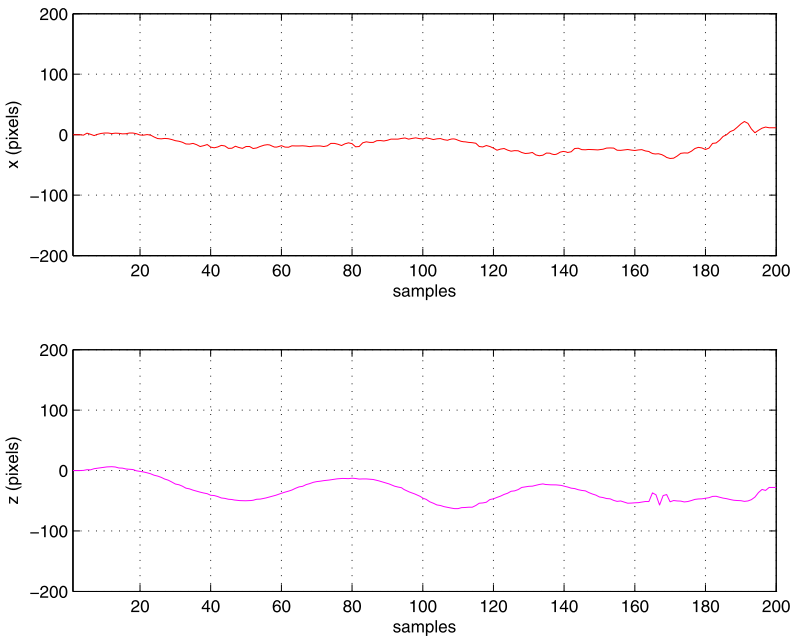
Fig. 6.18 Velocity regulation: Position results
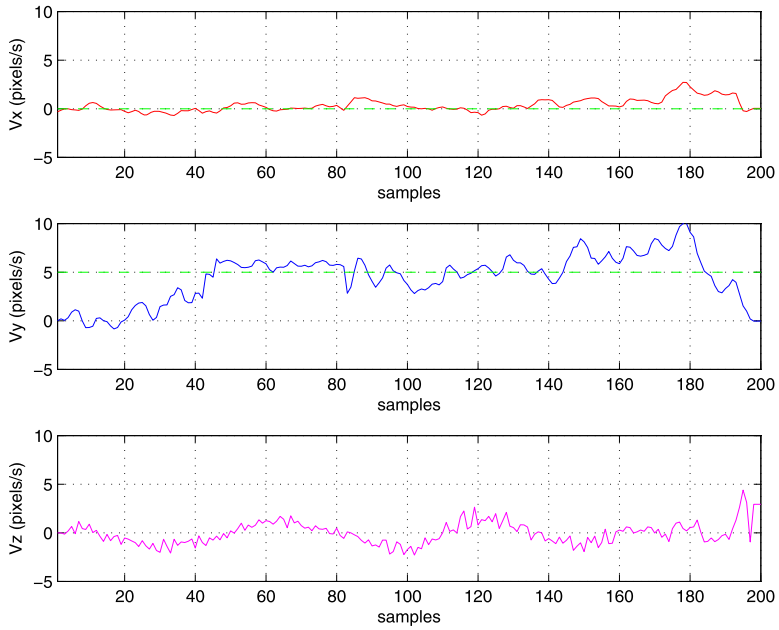
Fig. 6.19   Velocity regulation: Velocities results
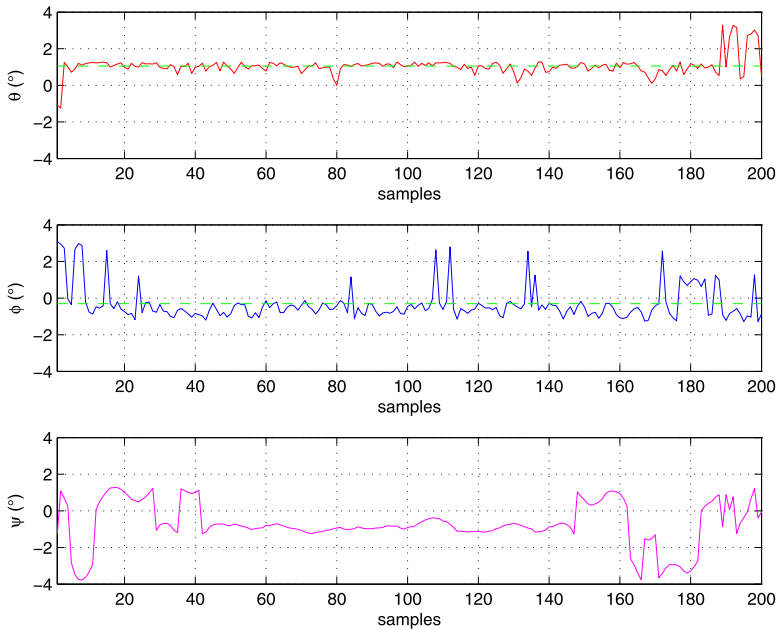
Fig. 6.20   Velocity regulation: Euler angles results

6.3.6 Final Comments

In this section, a vision-based strategy for 3-dimensional position control and ve-
locity regulation was proposed and tested in real-time applications. Using the UAV
quad-rotor platform, the proposed controller and vision algorithm were tested on a
mission consisting on road-following. The estimated information proved to be rich
enough to allow performing navigation missions. It has been shown that the vision-
based controller does not cause instability in the vehicle attitude.

6.4 Concluding Remarks

In this chapter, two different vision-based strategies for stabilizing a quad-rotor UAV
during flight were proposed and tested in real-time applications.

The first strategy consists of an homography estimation technique for extracting
the 3-dimensional position, as well as of an optical flow computation for deriv-
ing the vehicle's translational velocities. A technique for predicting the landing pad
position when its detection fails was presented also. The experimental results val-
idate the effectiveness of such approach. In a second phase, three different control
strategies were implemented over the same experimental set-up, with the purpose of
identifying the most effective controller for stabilizing the vehicle when using vi-
sual feedback. The second vision-based strategy was developed for controlling the
helicopter's 3-dimensional position, as well as achieving a velocity regulation. The
proposed controller and the vision algorithm were tested on a mission consisting
on road-following. The estimated information proved to be rich enough to allow
performing navigation missions. It has been shown that the vision-based controller
does not cause instability in the vehicle attitude.

From the set of experiments performed, it was found that the most difficult task
corresponds to the altitude stabilization. Indeed, the delay between the dynamics and
the visual estimation could cause the system to oscillate. Moreover, since the battery
is continuously consuming when the rotorcraft flies, the system power decreases
while the time passes. Thus, the rotorcraft tends to fall more easily than it rises.
This problem can be solved by making the altitude control asymmetric, providing
more importance to the positive control, which deals with the ascension, over the
negative control which allows the descent of the rotorcraft. With this arrangement,
the negative control is saturated faster, causing the rotorcraft to descend slowly when
it overruns the desired altitude.

A major disadvantage of both vision-based strategies is that, for a proper func-
tioning, the artificial landmarks must be completely detected by the imaging sensor.
For the first strategy, this requirement must be fulfilled keeping the landing pad un-
der the field of view of the camera. Concerning the second approach, the total width
of the road model needs to be under the field of view of the camera, in order to
estimate the variable $\frac{1}{\omega}$.

Aiming at overcoming the issues generated by the limitations of the monocular imaging sensors, a stereo vision system could be implemented, since it allows to directly estimate the relative 3-dimensional position of the imaging system with respect to its surrounding environment. The next chapter is devoted to the implementation of a stereo imaging system for the quad-rotor.

Chapter 7
Combining Stereo Imaging, Inertial and Altitude Sensing Systems for the Quad-Rotor

From previous chapters it is evident that, although monocular systems are low-weight designs and allow performing in an acceptable manner several vision-based tasks, they suffer from some drawbacks. One major inconvenience is that, in order to estimate its relative 3-dimensional position, they require using artificial markers of known dimensions. In addition, those markers must be inside the camera's field of view in every instant of time, which is commonly very hard to accomplish. A very promising solution to overcome those issues consists of the implementation of stereo vision systems and complementary sensors, which allow using the surrounding natural environment in order to estimate relative position.

In this chapter, a quad-rotor robotic platform equipped with a stereo imaging, inertial and altitude sensing system is presented. The objective of this research consists of enabling the UAV to autonomously perform take-off, relative positioning, navigation and landing, when evolving in unstructured, indoors, and GPS-denied environments.

The chapter is divided as follows. Section 7.1 deals with the problem of estimating the relative motion of a quad-rotor UAV in all six degrees of freedom. Aiming at this objective, an imaging, inertial, and altitude sensing system is introduced, which allows computing the 3-dimensional position and translational velocity of the vehicle with respect to its surrounding environment. The techniques for estimating the vehicle ego-motion and performing sensors fusion are presented in detail. Some experimental results consisting of motion estimation show the effectiveness of the proposed approach. With the purpose of identifying the most effective approach for combining visual odometry with inertial measurements, a Luenberger observer, a Kalman filter and a complementary filter are studied and compared in Sect. 7.2. To evaluate the performance of each strategy, real-time experiments consisting of motion detection and autonomous relative positioning are shown. Finally, some concluding remarks are presented in Sect. 7.3.

7.1 Estimating Motion

This Section presents the development of an imaging, inertial and altitude sensing system for the "Improved X-Flyer" quad-rotor. By using the images provided by a

pair of cameras, a visual odometry algorithm allows computing the ego-motion of the vehicle with respect to its surrounding environment. With the purpose of estimating the vehicle's 3-dimensional position and translational velocity in an accurate way, vision-based position estimation is combined with IMU-based accelerations and ultrasonic-based altitude measurements in a state estimator strategy.

7.1.1 Introduction

A basic requirement for UAVs consists of robust autonomous positioning and navigation. Information concerning the angular behavior of the aircraft are commonly estimated by means of inertial sensors (gyro, accelerometer, IMU), while the most common approach for estimating the 3-dimensional position and translational velocity is based on a GPS, which can be used in most of outdoor environments. In recent years, interest on the development of UAVs that can operate indoor or in urban environments has increased [41, 46, 84], thereby enabling an even wider range of robotic tasks to be accomplished. Unfortunately, when UAVs are required to navigate through urban environments or indoors, GPS signals may be noisy or even unavailable, leading to a poor or undesirable performance of the vehicle. Previous situations have motivated the search for alternative sensing solutions.

Fusion of data provided by a group of complementary sensors has proved to be an appropriate approach for fully estimating the state variables of a vehicle, as well as for sensing its surrounding environment. However, when working with a mini-UAV like the quad-rotor helicopter, the group of sensors and embedded electronics should be carefully chosen due to payload limitations. Taking into account the limitations just mentioned, the "Improved X-Flyer" robotic helicopter presented in Sect. 3.4.3 has been equipped with an IMU that provides the vehicle's attitude, as well as with three analog rate sensors to measure the platform angular rate. To estimate the helicopter's altitude at low and high flights, an ultrasonic sensor and an atmospheric pressure sensor are also installed. Attitude and position signals are read by an embedded micro-processor, which computes the control input that stabilizes the vehicle during flight. This basic sensor suit has been enhanced by including a stereo vision system in combination with a secondary IMU, with the purpose of exploiting complementary characteristics of imaging and inertial sensors.

Fast changes in angular rotation rates and linear accelerations are accurately measured by inertial sensors. Unfortunately, integration of these signals leads to unbounded low-frequency drift, making autonomous positioning and navigation a complex task. On the other hand, vision-based motion estimation provides good accuracy when the camera's field of view changes relatively slowly. In addition to the inertial and imaging sensors, an ultrasonic range finder is used to provide an estimation of the helicopter altitude. Performing a fusion of the three sensors data, each output information can be used to compensate for the weaknesses inherent in the others. For these studies, the helicopter motion is estimated by using visual odometry, which consists of tracking triangulated natural landmarks across sequential

Fig. 7.1 The quad-rotor robotic platform equipped with a stereo imaging, inertial, and altitude sensing system

stereo image pairs. Vision-based measurements are then combined with inertial and altitude sensor signals in a state observer, in order to provide accurate estimates of the UAV's (x, y, z) positions and $(\dot{x}, \dot{y}, \dot{z})$ translational velocities. The "Improved X-Flyer" aerial vehicle equipped with the sensing system just described is shown in Fig. 7.1.

7.1.1.1 Related Work

The process of incrementally estimating changes in robot pose by identifying and tracking visual landmarks in the environment is known as visual odometry (VO). Robotic vehicles are equipped with vision systems consisting of monocular [24], omnidirectional [31] and spherical cameras [55] in order to perform VO. An interesting result concerning vision-based localization of a UAV can be found in [13]. The authors implement a simultaneous localization and mapping algorithm (SLAM), using a monocular vision system. Their experimental results show accurate localization and positioning of a quad-rotor performing indoors. Stereo vision systems are also a common solution for estimating VO, since the depth of the tracked landmarks can be computed directly from known camera geometry. An example of systems equipped with VO based on stereo vision are the NASA Mars Exploration Rover (MER) robots [29]. During periods when wheel odometry is unreliable, such as when driving over high-slip terrain, the rovers position estimation rely only on visual estimates.

An example of combined visual and inertial sensing systems applied to aerial vehicles can be found in [30], where a vision and inertial sensing system is used for real-time control of a small helicopter. Height above the ground and optical flow due to ego-motion are provided by a pair of downward looking cameras. A complementary filter is used to fuse inertially and visually derived velocity information. In [51] the authors present a UAV navigation system which combines stereo VO with inertial measurements from an IMU. Vehicle position and attitude are obtained by fusing the motion estimates from both sensors in an extended Kalman filter. Information provided by the navigation system is analyzed off-line to evaluate the performance of a point to point navigation task.
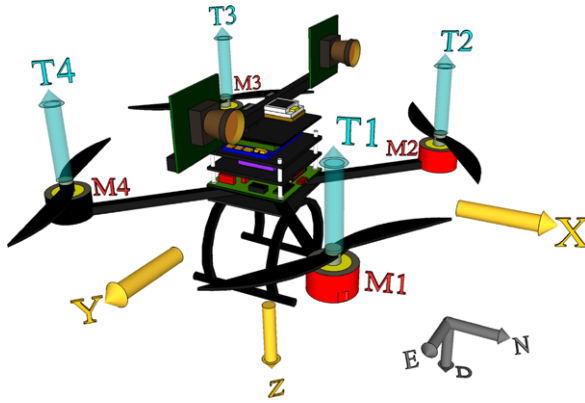
Fig. 7.2 NED diagram of the quad-rotor dynamical model

Unlike research work where experiments are performed over robotic platforms capable of carrying a considerable amount of payload (typically up to 5 kg), the present study focuses its efforts on the development of a small quad-rotor robotic helicopter equipped with a well suited sensing system. The research work presented here is closely related to aerial simultaneous localization and mapping (see, for example, [2]), although it is primarily concerned with the tasks of autonomous take-off, landing, positioning and point-to-point navigation, and so does not maintain a map of landmark positions.

7.1.2 System Setup

The dynamic model equations for an "X-Flyer" quad-rotor having two frontal motors and two rear motors is considered, see Fig. 7.2. Such model has been previously presented in Sect. 2.2.4, (2.61)–(2.66). The "X-Flyer" quad-rotor was considered for installing the proposed sensing system because under this arrangement the motors are positioned at 45° from the X and Y axes. When using a vision system pointing forwards, which is the case, it is desirable to have a clear view of the scene in front of the helicopter.

7.1.3 Experimental Platform Overview

Small quad-rotor UAVs have a maximum amount of vertical thrust that they can generate to remain airborne, which severely limits the amount of payload available for sensing and computation. This payload limitation restricts popular choices of embedded computers such as the PC-104, large-aperture cameras and high-fidelity

Fig. 7.3 Ascending
Technologies® Flying
Netbook mini-computer
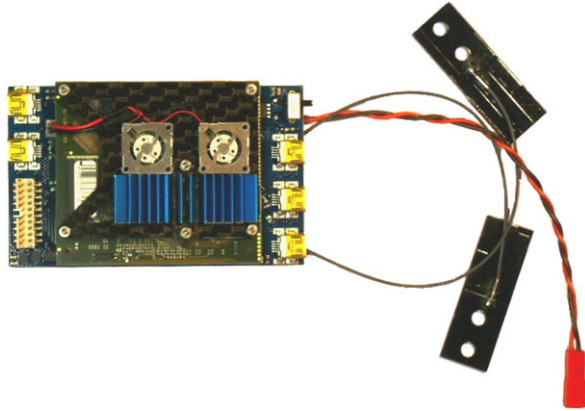
IMUs. Instead, small aerial robots rely on ultrasonic sensors, lightweight but lower-quality IMUs, and micro cameras having limited ranges and fields-of-view. It is common that the measurements provided by small sensors are noisier compared to their equivalents found in bigger robotics platforms.

The "Improved X-Flyer" quad-rotor helicopter is composed by a group of off-the-shelf components. Details concerning the components and electronics onboard were previously presented in Sect. 3.4.3. The robotic helicopter, as shown in Fig. 3.8, has a total weight of 945 grams. It has a payload capacity of 600 grams, which is enough for carrying the proposed sensing system described in the next section.

7.1.3.1   Visual and Inertial Navigation System

With the purpose of performing visual and inertial navigation studies, two monochromatic uEye cameras from IDS with a resolution of 376×240 pixels and a field of view of 75° were chosen. The cameras are mounted on a stereo bench with a 35 cm baseline and are placed facing forward. An additional 3DM-GX3 IMU from Microstrain® is mounted on the central section of the stereo bench with the purpose of measuring the acceleration of the vehicle. The cameras and the IMU are connected to an embedded Flying Netbook board developed by Ascending Technologies®, see Fig. 7.3. This mini-computer runs a program that deals with the next group of tasks. First, it synchronizes the uEye cameras, reads their images and compresses them in jpeg format. Secondly, it communicates with the 3DM-GX3 IMU and reads the acceleration data. The program generates a data buffer containing both images and acceleration values, and finally, this buffer is sent via 801.11n wireless communication to the supervisory ground station PC. Once the buffer is received, the images are decompressed and the acceleration values are read. Next, the images are processed by a VO estimation application, which runs in the 2.4 GHz ground station PC. Finally, VO estimation is combined with acceleration and altitude measurements in a state observer to provide accurate measurements of the

Fig. 7.4 Visual and inertial navigation system
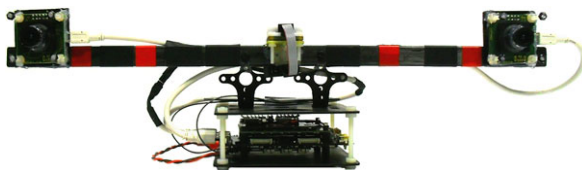
helicopter's (x, y, z) positions and $(\dot{x}, \dot{y}, \dot{z})$ translational velocities, which are required data for the control algorithm. The visual and inertial system can be seen in Fig. 7.4, and mounted at the top of the "Improved X-Flyer" quad-rotor helicopter in Fig. 7.1. The portable design of this system allows an easy installation–removal procedure from the robotic helicopter, without compromising the proper functioning of the platform.

7.1.3.2 Supervisory Ground Station

A supervisory ground station well suited for receiving sensor signals and executing imaging and control applications was developed. This system consists of a computer where a supervisory control application and an imaging processing application are executed. The computer has connected a flight simulator joystick, a bidirectional wireless link for data transmission and a 801.11n wireless link for receiving data sent by the Flying netbook. The wireless data link is used to receive and save information sent periodically by the aerial vehicle, which helps to characterize its performance during experiments, but also for sending a buffer containing estimated states and control commands to stabilize the helicopter's position. The control commands can be sent either by a user (by means of a flight joystick) or generated by the supervisory control application. The 801.11n wireless link receives real-time stereo images and acceleration measurements, which are processed by a computer vision program. The position and acceleration information extracted by this algorithm is placed on a fixed memory segment that is shared with the supervisory control application, where a state observer estimates the vehicles states required in the control algorithm.

The main function of the base station is the supervisory control application, it coordinates the running programs and allows to piloting the quad-rotor manually if needed. Data transmission between the supervisory ground station and the aerial vehicle is performed at a frequency of 13 Hz. Data sent from the base station to the UAV are prioritized since they carry information necessary for position control. The supervisory ground station is shown in Fig. 7.5. Figure 7.6 shows a schematic summarizing the steps performed by the Flying Netbook to obtain the images and acceleration values, it also shows the operations carried out in the supervisory ground station for obtaining the vehicle's states and sending the buffer with the required data for the controller.

Fig. 7.5 The supervisory ground station. From left to right: joystick, PC, 801.11n wireless link, XBEE09P data link
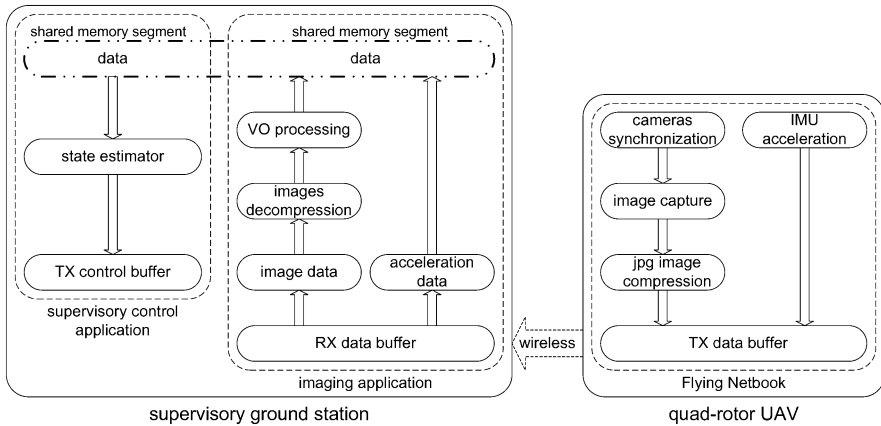
Fig. 7.6 A schematic of the process for estimating the vehicle's states

7.1.4 Stereo Visual Odometry

Stereo odometry is a vision-based technique that computes the ego-motion of a stereo rig through its surrounding environment by evaluating the camera's images. It can be thought of as a chain of several single subprocesses, where each of them relies on its predecessor's results. For each of these subprocesses, a variety of exchangeable methods are available [76].

The approach for performing stereo visual odometry is represented in Fig. 7.7. The algorithm begins in the previous left image, where features that can be tracked reliably are detected and identified. For each selected left image feature, its corresponding feature is searched in the previous right image. The 3-dimensional positions of the matched features pairs are reconstructed using triangulation. When a newer stereo images pair is available, features that have been reconstructed successfully are tracked from the previous left image to the current left image. Then, a similar correspondence and reconstruction step is performed for the current left image and current right image. At each time step, the process just described yields
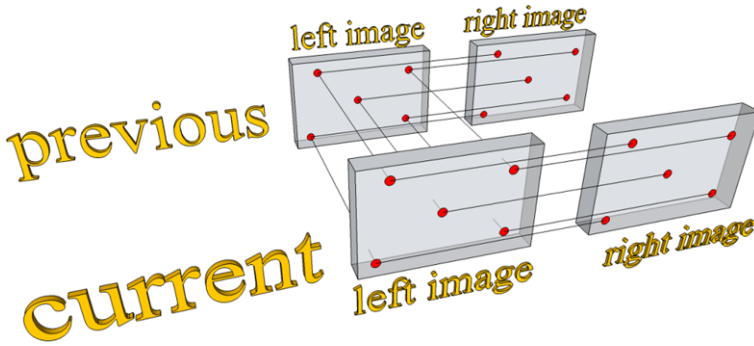
Fig. 7.7 Principle of the visual odometry algorithm

two sets of corresponding 3-dimensional features observations, before and after the helicopter platform has undergone an unknown rotation R and translation $\mathbf{T}$. Using the matched sets of features, the vehicle's relative motion in all six degrees of freedom can be computed.

7.1.4.1 Detecting Features

Finding interest features in the images is the first step towards stereo odometry. The algorithm proposed in [74] is being used to find salient characteristics in the left stereo image. In order to achieve a uniform detection of features over the entire image, four similar searching regions are defined (upper left, upper right, lower left, lower right). In addition, the parameters of the feature detector are adjusted to avoiding detection of features under some geometric proximity. The algorithm searches for a number of 150 candidate features over each searching region.

The pyramidal implementation of the Lucas–Kanade optical flow tracker [18] available in OpenCV [64] is used to track features between the left and right images, as well as from the previous to the current left frames. This algorithm allows tracking features robustly over large baselines, in addition, it is robust to the presence of image blur caused by motion. In order to validate feature correspondences between the left and right images, an error checking is performed at this stage. Based on the method available in [17], an accurate calibration of the stereo cameras has been performed and the obtained results were previously presented in Sect. 5.2.2. The calibration procedure allows the evaluation of the epipolar constraint

$$\mathbf{x}_r^{\mathrm{T}} F \mathbf{x}_l = 0 \pm S \tag{7.1}$$

which is a variation of (5.23) with an extra parameter S. In this formulation, $\mathbf{x}_*$ denotes the feature location in the respective (left or right) frame, F represents the fundamental matrix obtained from the extrinsic calibration of the stereo rig, and S is a threshold previously defined for acceptable noise level. It has been found experimentally that the vision algorithm has a good performance concerning

speed/accuracy if the number of successfully tracked features is maintained over 150. Aiming at maintaining this minimum number of successfully tracked features for every instant of time, the algorithm to find salient characteristics is executed each time the number of features is under 150.

7.1.4.2 3-dimensional Reconstruction

As soon as a pair of 2-dimensional features corresponding to a physical feature in space is found by the algorithm explained earlier, the position of this feature in 3-dimensional space can be reconstructed. A 3-dimensional point in space is projected into the left and right camera's image plane up to a scalar factor λ by [60]

$$\widehat{\mathbf{x}}_{il} = K_l \cdot \Pi_0 \cdot {}^oT_l \cdot \mathbf{X}_0 = \Pi_l \mathbf{X}_i \qquad (7.2)$$

$$\widehat{\mathbf{x}}_{ir} = K_r \cdot \Pi_0 \cdot {}^oT_l \cdot {}^lT_r \cdot \mathbf{X}_0 = \Pi_r \mathbf{X}_i \qquad (7.3)$$

where $\widehat{\mathbf{x}}_{il} = \lambda \mathbf{x}_{il}$ and $\widehat{\mathbf{x}}_{ir} = \lambda \mathbf{x}_{ir} \in \mathbb{R}^{3 \times 1}$ represent the projection of $\mathbf{X}_i \in \mathbb{R}^{4 \times 1}$ into the image plane of the left and right cameras through the projection matrices Π_l, $\Pi_r \in \mathbb{R}^{3 \times 4}$. Each one of these matrices consist of the intrinsic parameter matrix K_*, the standard projection matrix Π_0, as well as of the extrinsic parameters matrix ${}^lT_r$ representing the transformation of the right camera with respect to the left camera (obtained from the extrinsic stereo calibration presented in Sect. 5.2.2), and ${}^oT_l$ representing the transformation from the left camera with respect to the vehicle's center of gravity, which is simply expressed as

$$ {}^oT_l = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -17.5 \\ 0 & 0 & 1 & -10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (7.4)$$

From (7.2) and (7.3) a homogeneous system of three equations per camera is obtained. Only two of the three equations are linear independent since $\mathrm{rank}(\widehat{x}) = 2$. This leads to four constraints for the point $\mathbf{X}_i$ that must be recovered from both views:

$$\left(x_l \boldsymbol{\pi}_l^{3\mathrm{T}} - \boldsymbol{\pi}_l^{1\mathrm{T}}\right)\mathbf{X}_i = 0 \qquad (7.5)$$

$$\left(y_l \boldsymbol{\pi}_l^{3\mathrm{T}} - \boldsymbol{\pi}_l^{2\mathrm{T}}\right)\mathbf{X}_i = 0 \qquad (7.6)$$

$$\left(x_r \boldsymbol{\pi}_r^{3\mathrm{T}} - \boldsymbol{\pi}_r^{1\mathrm{T}}\right)\mathbf{X}_i = 0 \qquad (7.7)$$

$$\left(y_r \boldsymbol{\pi}_r^{3\mathrm{T}} - \boldsymbol{\pi}_r^{2\mathrm{T}}\right)\mathbf{X}_i = 0 \qquad (7.8)$$

where x_* and y_* denote the image coordinates of the feature and $\boldsymbol{\pi}_*^{j\mathrm{T}}$ denotes the jth row vector of the projection matrix $\Pi_* = [\boldsymbol{\pi}^1 \ \boldsymbol{\pi}^2 \ \boldsymbol{\pi}^3]^\mathrm{T}$. The set of equations (7.5)–(7.8) form a 4×4 homogeneous system that must be solved:

$$M\mathbf{X}_i = \mathbf{0} \qquad (7.9)$$

In practice the right-hand side of the above equation is different from zero due to inaccuracies and noise when finding correspondences or in the process of camera calibration. In other words, there is no intersection between the lines passing through each optical center of the cameras and $\mathbf{x}_l$–$\mathbf{x}_r$. Computing the eigenvalue decomposition of $M^T M$ a solution that minimizes the error $\|M\mathbf{X}_i\|$ of the homogeneous system in (7.9) in a least squares sense is obtained. The best solution for $\mathbf{X}_i$ is the eigenvector of $M^T M$ corresponding to the smallest eigenvalue. In order to complete the 3-dimensional reconstruction, $\mathbf{X}_i$ is normalized to make its last coordinate (X_4) equal to 1.

7.1.4.3 Estimating Motion

The procedure presented here follows the notation from [1]. At each time step, the reconstruction algorithm described above yields two sets of corresponding 3-dimensional features observations, before and after the helicopter stereo rig has undergone an unknown rotation R and translation $\mathbf{T}$. A method for computing the transformation between two sets of points is presented in [79]. This algorithm is intended to determine rotation R and translation $\mathbf{T}$ from the previous ($\mathbf{X}_p$) and the current ($\mathbf{X}_c$) set of points that minimize the mean square errors $e^2(R, \mathbf{T})$. The transformation can be described as

$$\mathbf{X}_{c,i} = R \cdot \mathbf{X}_{p,i} + \mathbf{T} \tag{7.10}$$

then

$$e^2(R, \mathbf{T}) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{X}_{c,i} - R \cdot \mathbf{X}_{p,i} - \mathbf{T}\|_2^2 \tag{7.11}$$

where $i = 1, \ldots, n$ denotes the number of features, and there must be at least three of them in order to uniquely estimate the transformation. First the rotation must be estimated, therefore the set of points are translated by their mean vectors $\overline{\mathbf{X}}_p$ and $\overline{\mathbf{X}}_c$ to the origin:

$$\overline{\mathbf{X}}_p = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_{p,i} \quad \tilde{\mathbf{X}}_{p,i} = \mathbf{X}_{p,i} - \overline{\mathbf{X}}_p \tag{7.12}$$

$$\overline{\mathbf{X}}_c = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_{c,i} \quad \tilde{\mathbf{X}}_{c,i} = \mathbf{X}_{c,i} - \overline{\mathbf{X}}_c \tag{7.13}$$

Using the translated set of points $\tilde{\mathbf{X}}_{p,i}$ and $\tilde{\mathbf{X}}_{c,i}$ a 3×3 covariance matrix can be computed as

$$\Sigma_{pc} = \frac{1}{n} \sum_{i=1}^{n} \tilde{\mathbf{X}}_{c,i} \cdot \tilde{\mathbf{X}}_{p,i}^T \tag{7.14}$$

R is obtained by performing the singular value decomposition (SVD) $\Sigma_{pc} = USV^{\mathrm{T}}$. Finally, to obtain as result a rotation and not a reflection $\det(U) \cdot \det(V)$ must be evaluated. If necessary, S must be corrected as

$$R = U\tilde{S}V^{\mathrm{T}} \tag{7.15}$$

$$\tilde{S} = \begin{cases} \mathrm{diag}(1, 1, 1) & \text{for } \det(U) \cdot \det(V) \geq 0 \\ \mathrm{diag}(1, 1, -1) & \text{for } \det(U) \cdot \det(V) < 0 \end{cases} \tag{7.16}$$

Once rotation is known, translation can be computed by inserting the two mean vectors $\overline{\mathbf{X}}_p$ and $\overline{\mathbf{X}}_c$ into (7.10):

$$\mathbf{T} = \overline{\mathbf{X}}_c - R\overline{\mathbf{X}}_p \tag{7.17}$$

which represents the transformation of two sets of points with respect to (w.r.t.) a fixed coordinate system. In the present case, the set of points is fixed while the stereo cameras (helicopter) coordinate system is moving. The rotation and translation of the helicopter to its previous body frame are obtained by

$$\Delta R = R^{\mathrm{T}} \tag{7.18}$$

$$\Delta \mathbf{T} = -R^{\mathrm{T}}\mathbf{T} \tag{7.19}$$

The Δ factor represents that the pose of the vehicle is only computed in the current time step w.r.t. the previous one. In order to estimate the pose of the vehicle w.r.t. its starting pose T_0, homogeneous transformations must be applied to perform a chaining of all of the Δ poses obtained:

$$T_{\text{current}} = \underbrace{T_0 \cdot \Delta T_{t-m+1} \cdot \ldots \cdot \Delta T_{t-1}}_{T_{\text{prevoius}}} \cdot \Delta T_t \tag{7.20}$$

with

$$T = \begin{bmatrix} R & \mathbf{T} \\ \underline{0} & 1 \end{bmatrix} \tag{7.21}$$

By performing a right-multiplication of the previous pose T_{previous} with the latest ΔT_t pose, the current motion w.r.t. the previous pose is added to the previous estimated pose. The pose where the vehicle started is represented by T_0, and, in the present case, it corresponds to the identity matrix. The group of last Δ poses can be stored in the single 4×4 matrix T_{previous}. Therefore, the pose update in every time step requires only one matrix multiplication. In order to describe the helicopter's attitude from VO, the (ψ, θ, ϕ) Euler angles must be extracted from the rotation matrix R as

$$\theta = \mathrm{atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{12}^2}\right) \tag{7.22}$$

$$\psi = \mathrm{atan2}\left(\frac{r_{21}}{\cos(\theta)}, \frac{r_{11}}{\cos(\theta)}\right) \tag{7.23}$$
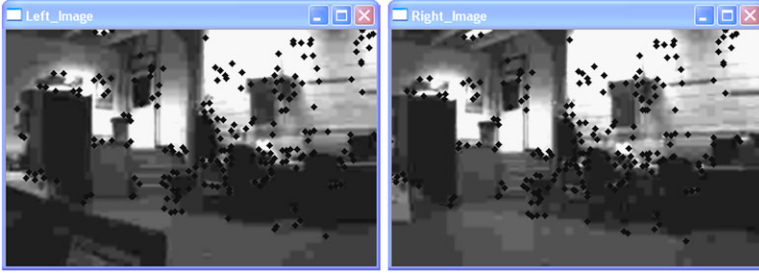
Fig. 7.8 The visual odometry algorithm running on the stereo images

$$\phi = \text{atan2}\left(\frac{r_{32}}{\cos(\theta)}, \frac{r_{33}}{\cos(\theta)}\right) \qquad (7.24)$$

where $r_{\text{row,column}}$ denotes a specific entry of R.

The least square method explained cannot be used without any further improvement to make it robust against outliers in the data. RanSaC [35] is a well known method for robust estimation. As in [62], it has been decided to implement the SVD algorithm as an hypothesis generator for the RanSaC procedure. Once the hypothesis with the maximum number of inliers is found, the solution is recomputed using all the inliers. An image of the visual odometry algorithm running on the stereo images is presented in Fig. 7.8. The scene corresponds to an unstructured indoors environment. The tracked features used for obtaining visual odometry are highlighted with black dots.

7.1.5 A Simple Strategy for Imaging, Inertial and Altitude Data Fusion

With the purpose of obtaining an estimate of the vehicle's 3-dimensional position and velocity, imaging, inertial and altitude measurements are fused in a Kalman filter. The state vector is defined as

$$\mathbf{x}[k] = [x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \quad z \quad \dot{z} \quad \ddot{z}]^{\text{T}} \qquad (7.25)$$

with (x, y, z) representing the position of the quad-rotor in the global NED frame. The observation vector is defined as

$$\mathbf{z}[k] = [x_{\text{vo}} \quad x_{a,\text{imu}} \quad y_{\text{vo}} \quad y_{a,\text{imu}} \quad z_{\text{vo}} \quad z_{a,\text{imu}} \quad z_{\text{as}}]^{\text{T}} \qquad (7.26)$$

where $(x_{\text{vo}}, y_{\text{vo}}, z_{\text{vo}})$ represents measurements of the helicopter position provided by the visual odometry algorithm, and $(x_{a,\text{imu}}, y_{a,\text{imu}}, z_{a,\text{imu}})$ are linear accelerations provided by the IMU. The measurement provided by the altitude sensor is represented by z_{as}. The filter fuses z_{vo} and z_{as} in order to compute a better estimation of the vehicle's altitude. The Kalman filter functions used in this experiment
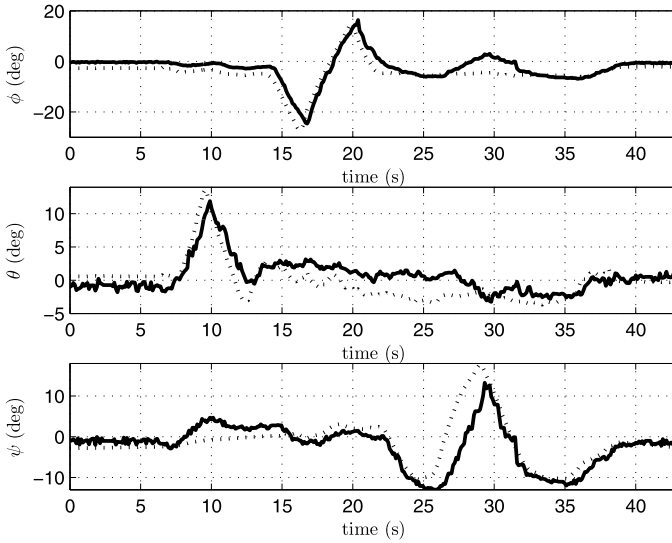
Fig. 7.9 Euler angles comparison: VO vs. IMU. *Solid lines* represent VO estimation, *dotted lines* represent IMU data

were provided by the OpenCV library, which allows performing all the tasks related to the states estimation and data fusion.

7.1.6 Experimental Results

Once the proposed sensing system was correctly installed onboard the UAV, several tests were conducted. It has been noticed that, in spite of the considerable amount of extra payload, the helicopter performance is not degraded. During such tests, the scene surrounding the helicopter was a simple unstructured indoor environment, like the one shown in Fig. 7.8.

The first experiment consisted of comparing the Euler angles estimated by VO versus the Euler angles provided by the Microbotics IMU, which are considered as the real orientation experienced by the quad-rotor. During this experiment, tilting movements of the helicopter were generated manually (by using the joystick). The Euler angles estimated by visual odometry and measured by the IMU are presented in Fig. 7.9. Next, a second experiment was conducted, consisting of a manually controlled flight of the quad-rotor over a trajectory forming a square of 150 cm × 300 cm, with a fixed altitude of 50 cm. The first movement performed consisted of 150 cm forward. Next, the quad-rotor was flown 300 cm to the right. At this point, a backwards displacement of 150 cm was performed. Finally, the quad-rotor was flown 300 cm to the left, in order to reach the same position at which it started. The system achieves an acceptable estimation of the movement described by the helicopter. The resulting estimated trajectory is presented in Fig. 7.10, this does not
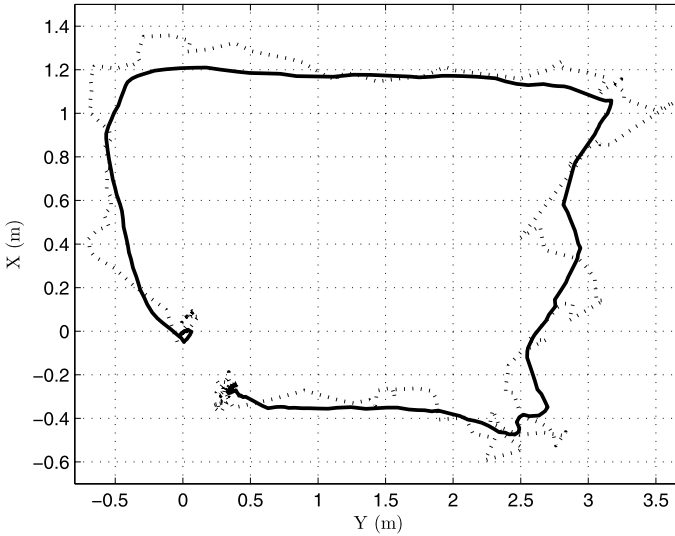
Fig. 7.10 Path that the helicopter has flown. This does not represent ground truth, it is the position estimated by the algorithm. The *dotted line* represents VO alone, *solid line* represents VO + IMU estimation

represent ground truth, it is the position estimated by the algorithm. The trajectory of 900 cm was performed in 45 seconds of flight approximatively.

Finally, a third experiment was performed, consisting of a manually controlled flight of the quad-rotor, where the helicopter altitude was varied from 50 cm to 200 cm. The idea of performing this test consists of verifying how reliable is the combination of imaging, inertial and altitude sensing system for estimating the quad-rotor altitude. Results of this experiment are shown in Fig. 7.11. The altitude estimated by means of the data fusion (continuous line) presents a similar behavior to the altitude estimated by the ultrasonic sensor (dotted line), however, still some differences exist between them. It is supposed that differences are induced by outliers in the VO algorithm, and we are still working to solve such issues.

During the previous experiments, data from the proposed sensing system were estimated at a rate of 13 Hz. From earlier work concerning quad-rotor applications (see for example [37]), it is known that such working frequency is appropriate for position control purposes. Real-time experiments consisting of autonomous hover and manually controlled flight of the quad-rotor can be seen in http://www.youtube.com/watch?v=IbTCfq_m7wc.

7.1.7 Final Comments

This section presented the development of a quad-rotor UAV equipped with a visual, inertial and altitude sensing system, which enables the helicopter to fully estimate
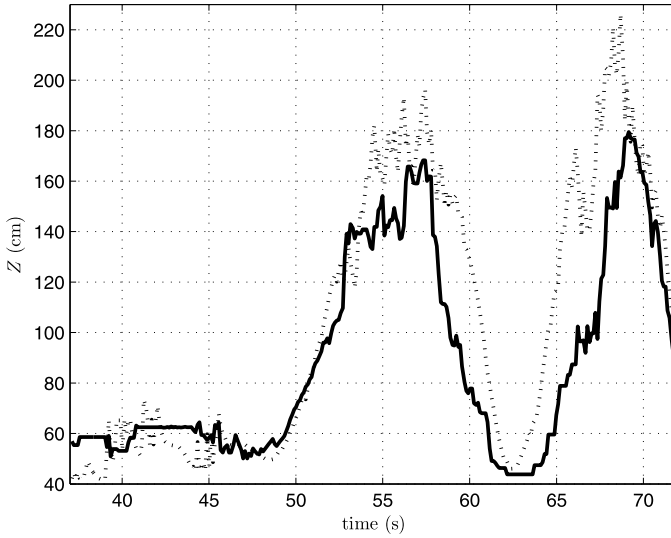
Fig. 7.11 Real-time altitude estimation. Ultrasonic sensor (*dotted line*), imaging, inertial, and altitude combination (*continuous line*)

its states without using GPS or artificial visual landmarks. In the proposed approach, a Kalman filter provided by the OpenCV library was implemented to combine VO measurements, IMU acceleration data and altitude sensor signals, with the purpose of providing accurate estimations of the states describing the behavior of the platform. The system performance was tested indoors, under real-time flight conditions.

Experimental results obtained during manually controlled flights have shown that the proposed system is capable of determining the quad-rotor 3-dimensional position and translational velocities in an accurate way. Such information is estimated at a rate of 13 Hz, which is adequate for real-time control purposes.

7.2   Comparison of Different State Estimation Algorithms for Quad-Rotor Control

The quad-rotor control requires knowledge of the state of the aircraft, mainly linear and angular velocity and position. Normally, miniature UAVs payload is severely restricted to avoid unnecessary energy consumption and increase autonomy. Therefore there is a limitation on the number and the weight of onboard sensors. As a consequence, it is crucial to develop efficient state observers for UAVs. In fact, in most of the literature on small flying robots control, the state is assumed to be completely available, see for example [27]. Standard onboard inertial sensors for UAVs are accelerometers and gyros which measure angular velocity. Small size GPS can be used to estimate the position and velocity of a UAV, however, the measurements

are reliable only when the sensor is far from urban areas. In this section, how to efficiently combine inertial measurements and imaging sensors is studied, with the purpose of effectively estimating the states of a UAV.

7.2.1 Introduction

Several research works deal with estimating the states of an aircraft. A method for obtaining the velocity of a quad-rotor UAV from acceleration measurements is presented in [12]. This approach, based on an adaptive observer technique, is compared with an extended Kalman filter in [11]. Although very promising, this technique leads to a high order observer with high computational requirements that is not suited for real-time experimental applications. Following a similar approach, a state observer of order equal to the dimension of the state vector is presented in [19]. The estimator's performance, presented in simulations, suggest that this approach can be applied in real-time applications. In [67], the authors propose a vision-based method to compute the translational speed, in the $X-Y$ plane, of a UAV equipped with eight rotors. Similarly, a vision-based optical flow method was implemented in [37] for the estimation of the translational velocity of a quad-rotor. The two previous approaches have been validated with real-time experiments consisting of stabilized hover flights. A different approach, consisting of fusion of vision and inertial measurements is presented in [22] and [30]. This method, based on complementary filtering, provides an estimation of the vertical position and velocity of a robotic helicopter of 8 kg and a length of 1.8 m. A realistic approach for the estimation of position and velocity of a fixed-wing UAV equipped with infrared camera and inertial sensors is presented in [43]. This method is suitable to deal with instants of time when the vision system is unable to provide accurate data.

Unlike previous work which considers acceleration or vision measurements for estimating velocity, the objective here is including both visual and inertial information in the estimators. Furthermore, the implementation of such estimators is intended for a quad-rotor mini-UAV platform. Taking into account the restrictions imposed by the payload capacity and power consumption of a small UAV, the "Improved X-Flyer" quad-rotor shown in Fig. 7.12 has been equipped with an imaging and inertial sensing system consisting of a stereo rig and an IMU. These passive sensors installed onboard can provide the relative position of the quad-rotor with respect to an inertial frame, as well as the translational acceleration experienced by the vehicle. The proposed sensing system provides an appropriate framework for the development of state observers and complementary filters that combine visual and inertial information, in order to provide an accurate estimation of the position and velocity of the quad-rotor.

The goal of this research consists of identifying the most effective approach for the combination of imaging and inertial information, in order to obtain an accurate estimation of the translational velocity of the helicopter. Three of the most commonly used state estimators found in the literature have been chosen: Luenberger

Fig. 7.12 "Improved
X-Flyer" equipped with
imaging and inertial sensors
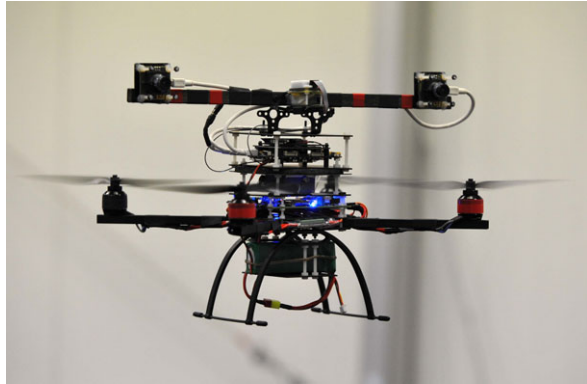
observer, Kalman filter and complementary filtering. The estimators performance
was evaluated in real-time experiments, consisting of hover flight and position sta-
bilization of the quad-rotor in unstructured indoors environments. Experimental re-
sults have shown that, even though the three methodologies achieve an acceptable
estimation of the vehicle's position and velocity, the helicopter controller reacts dif-
ferently according to the applied estimator.

7.2.2 Problem Statement

Consider a quad-rotor hovering in an indoor unstructured environment. The quad-
rotor is equipped with an IMU and analog rate gyros that directly measures the
Euler angles and the angular rates, respectively. Such information can be applied in
a control strategy for stabilizing the platform attitude. The vehicle is also equipped
with a portable imaging and inertial sensing system. The visual part provides the
3-dimensional position of the helicopter with respect to a fixed reference frame (see
Sect. 7.1.4), while the inertial part of the system provides the acceleration of the
vehicle's center of mass. The objective of this study consists of using the informa-
tion provided by both the imaging and inertial sensing system, in three different
approaches to obtain the most accurate estimate of the helicopter translational ve-
locity. The UAV communicates with the supervisory ground station, see Fig. 7.5,
where the sensors fusion takes place.

7.2.2.1 Measurement Model

Taking into account the measured variables, and given that the stereo cameras and
IMU move together as a rigid body, the discrete-time measurement model can be
written as [57]

$$\mathbf{x}[k+1] = A\mathbf{x}[k] \qquad\qquad (7.27)$$

$$\mathbf{y}[k] = C\mathbf{x}[k] \tag{7.28}$$

where the system states and outputs are represented by

$$\mathbf{x}[k] = [x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \quad z \quad \dot{z} \quad \ddot{z}]^{\mathrm{T}} \tag{7.29}$$

$$\mathbf{y}[k] = [x \quad \ddot{x} \quad y \quad \ddot{y} \quad z \quad \ddot{z}]^{\mathrm{T}} \tag{7.30}$$

The state matrix of the measurement model is

$$A = \begin{bmatrix} A_x & 0_A & 0_A \\ 0_A & A_y & 0_A \\ 0_A & 0_A & A_z \end{bmatrix} \tag{7.31}$$

where

$$A_{x,y,z} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}; \qquad 0_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The output matrix of the measurement model is

$$C[k] = \begin{bmatrix} C_x & 0_C & 0_C \\ 0_C & C_y & 0_C \\ 0_C & 0_C & C_z \end{bmatrix} \tag{7.32}$$

where

$$C_{x,y,z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \qquad 0_C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The available measurements in the above model are the position (x, y, z) and the acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$. Using these measurements in a state observer, an estimation of the translational velocity can be computed. For the construction of the state observers, the model must satisfy the observability condition. This is satisfied since the system observability matrix is full rank.

7.2.3 Design of Imaging-Inertial State Observers

This section presents the development of three state observers for the estimation of the quad-rotor translational velocity. The estimation is obtained by fusing imaging and inertial data.

7.2.3.1 Luenberger State Observer

The proposed state observer for the model (7.27)–(7.28) is a classical discrete-time Luenberger observer [28]. The variables of the state observer are denoted by $\hat{\mathbf{x}}$ and

$\hat{\mathbf{y}}$ and are defined by

$$\hat{\mathbf{x}}[k+1] = A\hat{\mathbf{x}}[k] - L(\hat{\mathbf{y}}[k] - \mathbf{y}[k]) \tag{7.33}$$

$$\mathbf{y}[k] = C\hat{\mathbf{x}}[k] \tag{7.34}$$

The L matrix of the observer is defined as

$$L = \begin{bmatrix} L_x & 0_L & 0_L \\ 0_L & L_y & 0_L \\ 0_L & 0_L & L_z \end{bmatrix}$$

where

$$L_{x,y,z} = \begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \\ l_5 & l_6 \end{bmatrix}; \qquad 0_L = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The observer is asymptotically stable if the error $e[k] = \hat{\mathbf{x}}[k] - \mathbf{x}[k]$ converges to zero when $k \to \infty$. The error for a Luenberger observer satisfies $e[k+1] = (A - LC)e[k]$. Choosing the eigenvalues such that the matrix $(A - LC)$ is Hurwitz, the Luenberger observer for this discrete-time system is therefore asymptotically stable. Finally, the equations describing state estimations are denoted by

$$\hat{x}[k+1] = \hat{x}[k] + T\dot{\hat{x}}[k] + \frac{T^2}{2}\ddot{\hat{x}}[k] - l_1(\hat{x}[k] - x[k])$$
$$- l_2(\ddot{\hat{x}}[k] - \ddot{x}[k]) \tag{7.35}$$

$$\dot{\hat{x}}[k+1] = \dot{\hat{x}}[k] + T\ddot{\hat{x}}[k] - l_3(\hat{x}[k] - x[k])$$
$$- l_4(\ddot{\hat{x}}[k] - \ddot{x}[k]) \tag{7.36}$$

$$\ddot{\hat{x}}[k+1] = \ddot{\hat{x}}[k] - l_5(\hat{x}[k] - x[k]) - l_6(\ddot{\hat{x}}[k] - \ddot{x}[k]) \tag{7.37}$$

$$\hat{y}[k+1] = \hat{y}[k] + T\dot{\hat{y}}[k] + \frac{T^2}{2}\ddot{\hat{y}}[k] - l_1(\hat{y}[k] - y[k])$$
$$- l_2(\ddot{\hat{y}}[k] - \ddot{y}[k]) \tag{7.38}$$

$$\dot{\hat{y}}[k+1] = \dot{\hat{y}}[k] + T\ddot{\hat{y}}[k] - l_3(\hat{y}[k] - y[k])$$
$$- l_4(\ddot{\hat{y}}[k] - \ddot{y}[k]) \tag{7.39}$$

$$\ddot{\hat{y}}[k+1] = \ddot{\hat{y}}[k] - l_5(\hat{y}[k] - y[k]) - l_6(\ddot{\hat{y}}[k] - \ddot{y}[k]) \tag{7.40}$$

$$\hat{z}[k+1] = \hat{z}[k] + T\dot{\hat{z}}[k] + \frac{T^2}{2}\ddot{\hat{z}}[k] - l_1(\hat{z}[k] - z[k])$$
$$- l_2(\ddot{\hat{z}}[k] - \ddot{z}[k]) \tag{7.41}$$

$$\dot{\hat{z}}[k+1] = \dot{\hat{z}}[k] + T\ddot{\hat{z}}[k] - l_3(\hat{z}[k] - z[k])$$

$$- l_4(\dot{\hat{z}}[k] - \dot{z}[k]) \tag{7.42}$$

$$\ddot{\hat{z}}[k+1] = \ddot{\hat{z}}[k] - l_5(\hat{z}[k] - z[k]) - l_6(\ddot{\hat{z}}[k] - \ddot{z}[k]) \tag{7.43}$$

where $\dot{\hat{x}}$, $\dot{\hat{y}}$ and $\dot{\hat{z}}$, represent the estimated velocities, required in the quad-rotor control strategy.

7.2.3.2 Kalman Filter

The second approach for combining vision and inertial data is the discrete Kalman filter. Let us rewrite the measurement model (7.27)–(7.28) in the form

$$\mathbf{x}[k+1] = \Phi\mathbf{x}[k] + \mathbf{w}[k] \tag{7.44}$$

$$\mathbf{z}[k] = H\mathbf{x}[k] + \mathbf{v}[k] \tag{7.45}$$

where $\mathbf{x} \in \mathbb{R}^{9\times1}$ (equation (7.29)) is the state vector, $\Phi \in \mathbb{R}^{9\times9}$ is the state transition matrix, $\mathbf{w} \in \mathbb{R}^{9\times1}$ is a white sequence having known covariance structure, $\mathbf{z} \in \mathbb{R}^{6\times1}$ is the measurement vector, $H \in \mathbb{R}^{6\times9}$ is a matrix giving the ideal connection between the measurement and the state vector, $\mathbf{v} \in \mathbb{R}^{6\times1}$ is the measurement error assumed to be a white sequence with known covariance structure.

The first part of the Kalman process consists of assuming one has an *a priori* estimate of the states, denoted as

$$\hat{\mathbf{x}}[k]^- = \Phi\hat{\mathbf{x}}[k-1] + \mathbf{w}[k]$$

Let us also assume knowledge of the error covariance matrix associated with $\hat{\mathbf{x}}[k]^-$, denoted as

$$P[k]^- = \Phi P[k-1]\Phi^{\mathrm{T}} + Q[k-1]$$

where Q is the covariance matrix for $\mathbf{w}$. With the assumption of the prior estimate, one seeks to use the measurements in $\mathbf{z}[k]$ to improve the prior estimate. Let us choose a linear blending of the noisy measurement and the *a priori* estimate as follows:

$$\hat{\mathbf{x}}[k] = \hat{\mathbf{x}}[k]^- + K[k](\mathbf{z}[k] - H\hat{\mathbf{x}}[k]^-)$$

where $\mathbf{x}[k]$ is the updated estimate and $K[k]$ is the blending factor, usually known as the *Kalman gain*. A particular $K[k]$ must be identified in order to obtain an update estimate that is optimal in some sense. The solution for the optimal gain can be expressed as

$$K[k] = P[k]^- H^{\mathrm{T}}(HP[k]^- H^{\mathrm{T}} + R[\mathbf{k}])^{-1}$$

where R is the covariance matrix for $\mathbf{v}$. The covariance matrix associated with the optimal estimate may now be computed as

$$P[k] = (I - K[k]H)P[k]^-$$

Finally, the updated estimated $\hat{\mathbf{x}}$ is projected ahead using the transition matrix

$$\hat{\mathbf{x}}[k+1]^- = \Phi\hat{\mathbf{x}}[k]$$

with its corresponding covariance matrix

$$P[k+1]^- = \Phi P[k]\Phi^{\mathrm{T}} + Q[k]$$

This process yields the required values at time $[k+1]$, and the measurement $\mathbf{z}[k+1]$ can be assimilated just as in the previous step. Further details can be found in [20] and [21].

7.2.3.3 Complementary Filter

The complementary filtering technique is the third approach used for data fusion. The definition of a complementary filter refers to the use of two or more transfer functions, which are mathematical complements of one another. For the present studies, the position estimated by means of the stereo vision system is combined with the IMU acceleration measurements in order to estimate the quad-rotor velocities $\dot{\hat{\mathbf{x}}} = (\dot{x}, \dot{y}, \dot{z})$. Let us use the next structure of the complementary filter:

$$\dot{\hat{\mathbf{x}}} = G(s)\hat{\mathbf{x}}_D + (1 - G(s))\hat{\mathbf{x}}_I \tag{7.46}$$

where $\hat{\mathbf{x}}_D = (\dot{x}_D, \dot{y}_D, \dot{z}_D)$ are velocity estimates obtained by directly differentiating the position obtained from the imaging system, and $\hat{\mathbf{x}}_I = (\dot{x}_I, \dot{y}_I, \dot{z}_I)$ are velocities obtained by integrating the translational acceleration data. $G(s)$ is typically a low-pass filter of the form

$$G(s) = \frac{\alpha}{s + \alpha}$$

and its complement

$$1 - G(s) = \frac{s}{s + \alpha}$$

would be a high-pass filter. This is advantageous in the case where the estimate multiplying the low-pass filter is a low-bandwidth sensor, and the measurement multiplying the high-pass filter has a DC offset. Defining $G_1(s) = G(s)$ and $G_2(s) = 1 - G(s)$, a discretized expression for the filters can be expressed as

$$G_1(z) = \frac{(1 - e^{-\frac{T}{\tau}})z^{-1}}{1 - e^{-\frac{T}{\tau}}z - 1}; \qquad G_2(z) = \frac{1 - z^{-1}}{1 - e^{-\frac{T}{\tau}}z - 1} \tag{7.47}$$

where $\alpha = \frac{1}{\tau}$. $\tau = \frac{1}{w_c} = \frac{1}{2\pi f_c}$ represents the time constant of the filters, and f_c represents the filter's cutoff frequency.

In the filter implementation, the velocity obtained by integrating the acceleration is expressed as

$$\hat{\mathbf{x}}_I[k+1] = e^{-\frac{T}{\tau}}\hat{\mathbf{x}}_I[k] + \left(1 - e^{-\frac{T}{\tau}}\right)\mathbf{x}_I[k] \tag{7.48}$$

whereas the velocity obtained by differentiating the position is represented by

$$\hat{\mathbf{x}}_D[k+1] = e^{-\frac{T}{\tau}}\hat{\mathbf{x}}_D[k] + \left(\mathbf{x}[k] + \mathbf{x}[k-1]\right)T \tag{7.49}$$

Finally, using (7.48) and (7.49), the complementary filter expression is obtained,

$$\hat{\mathbf{x}}[k+1] = \hat{\mathbf{x}}_I[k+1] + \hat{\mathbf{x}}_D[k+1] \tag{7.50}$$

Further details of the complementary filter can be found in [30].

7.2.4 Experimental Results

Three real-time experiments were conducted, in order to verify the performance of the position control when using the velocity data provided by the state estimators. Such experiments are explained next.

The vehicle is flown manually until it reaches 1.5 meters above the ground. At this point, the current (x, y, z) quad-rotor 3-dimensional position, computed by means of the stereo vision algorithm is registered and such data are then used as the position reference during the rest of the experiment. The state estimators working frequency is 13 Hz, which is also the rate of the position control. The control strategy implemented during experiments is the nested saturations control presented in Sect. 6.2.3.1. The parameter values used for the altitude and the yaw controller are: $k_{pz} = 0.68$, $k_{vz} = 1.6$, $k_{p\psi} = 38$, $k_{p\psi} = 1350$. The parameters used for the nested saturations controller are: $b_4 = 0.4700$, $b_3 = 0.2349$, $b_2 = 0.1174$, and $b_1 = 0.0287$.

The first state estimator tested corresponds to the Luenberger observer. The Luenberger gains applied during the experiments are: $l_1 = 0.210$, $l_2 = 0.005$, $l_3 = 0.020$, $l_4 = 0.100$, $l_5 = 0.000$, $l_6 = 0.300$. The corresponding x and y positions as well as the z-error position described by the platform, respectively, are represented in Fig. 7.13. The velocity estimation for translational displacement is represented in Fig. 7.14. Finally, the Euler angles behavior during such experiment is represented in Fig. 7.15.

The second state estimator tested was the Kalman filter. The process and the measurement errors applied during the experiment were $\mathbf{w} = 0.150$ and $\mathbf{v} = 0.100$, respectively. The corresponding x and y positions as well as the z-error position described by the platform, respectively, are represented in Fig. 7.16. The velocity estimation for translational displacement is represented in Fig. 7.17. Finally, the Euler angles behavior during such experiment is represented in Fig. 7.18.
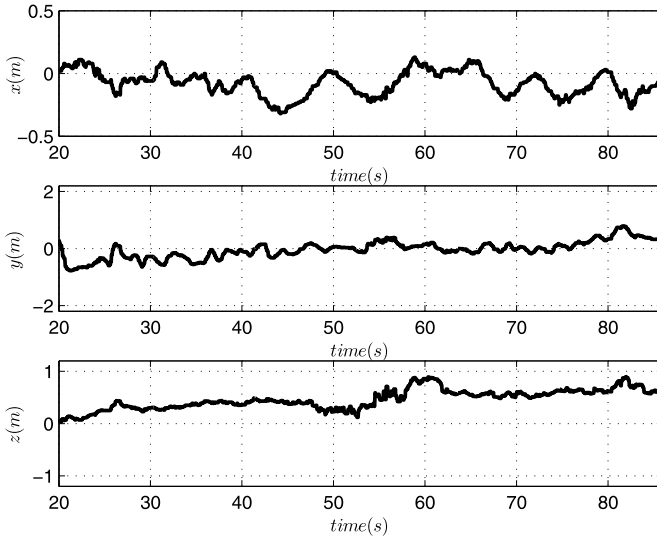
Fig. 7.13 Luenberger observer approach: $x-y$ positions and z-error position
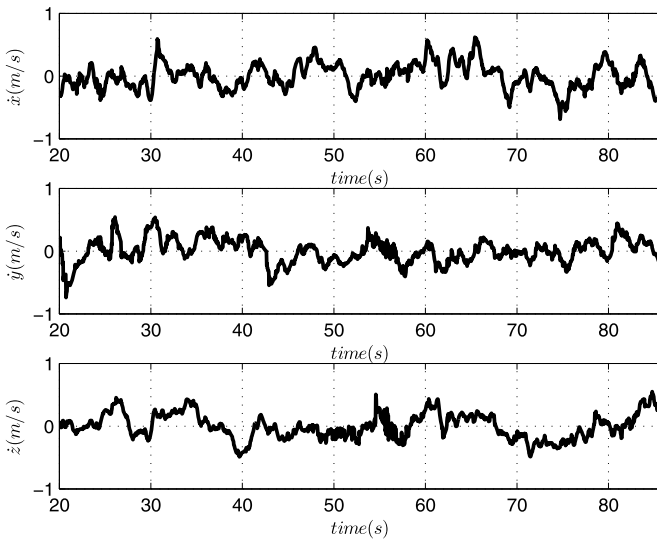
Fig. 7.14 Luenberger observer approach: $x-y$ and z translational velocities

The third state estimator tested was the complementary filter. The corresponding x and y positions as well as the z-error position described by the platform, respectively, are shown in Fig. 7.19. The velocity estimation for translational displacement is shown in Fig. 7.20. Finally, the Euler angle behavior during such experiment is shown in Fig. 7.21.
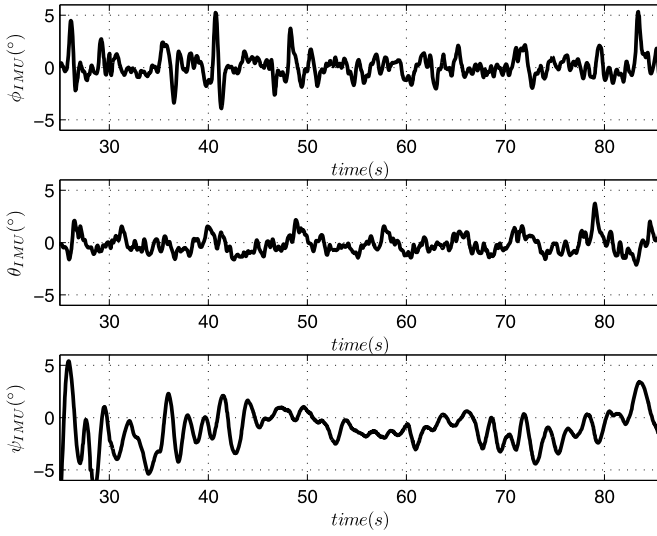
Fig. 7.15 Luenberger observer approach: Behavior of the Euler angles
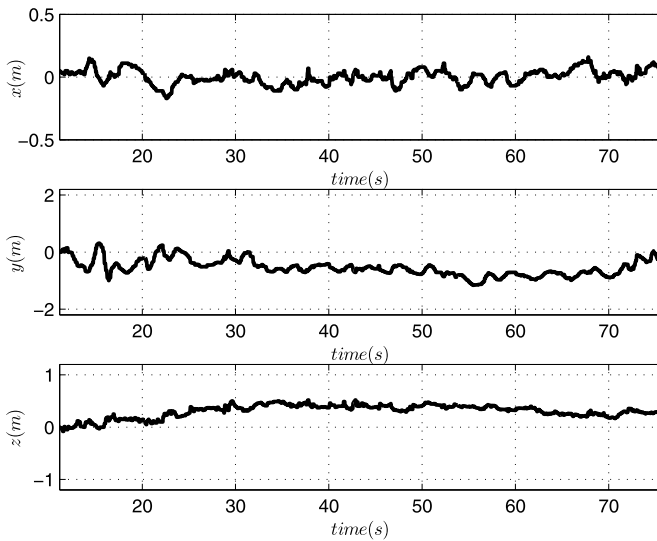
Fig. 7.16 Kalman filter approach: $x-y$ positions and z-error position

The estimated positions of the vehicle have been analyzed, which are shown in Figs. 7.13, 7.16, and 7.19, as well as the angular behavior shown in Figs. 7.15, 7.18, and 7.21 in order to detect how the quad-rotor position control is influenced by the performance of the velocity estimators.
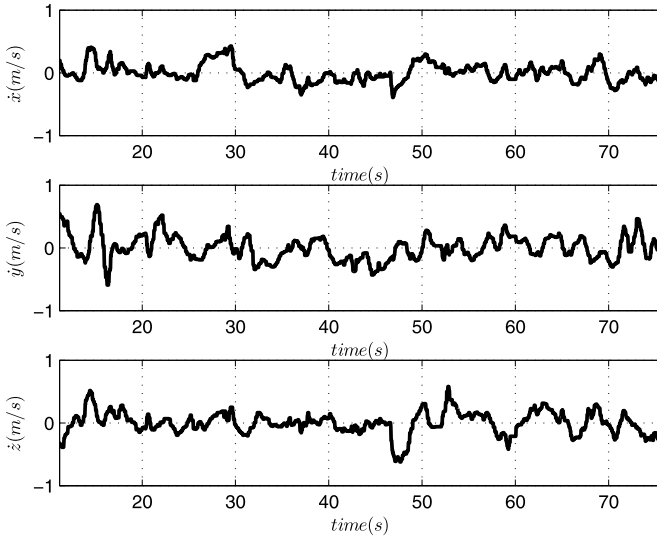
Fig. 7.17   Kalman filter approach: $x-y$ and z translational velocities
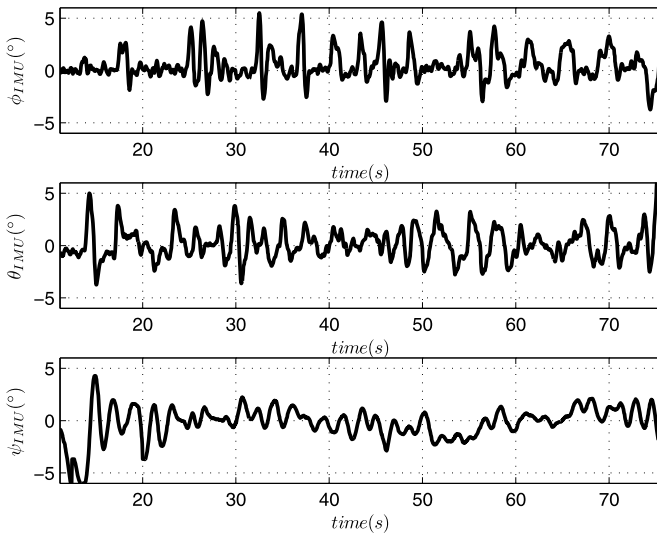
Fig. 7.18   Kalman filter approach: Behavior of the Euler angles

The velocity computed by the Luenberger observer, see Fig. 7.14, is the less-smoother signal if compared with the response of the other two estimators, it is considered that this degrades the performance of the helicopter controller. Figure 7.13 shows that the controller makes the helicopter change its x position from -0.3 m to 0.2 m. In addition, the graph shows that the quality of the altitude controller is
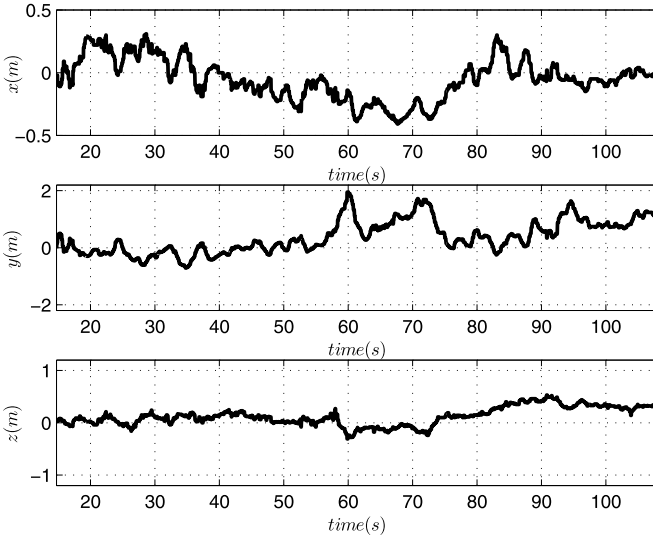
Fig. 7.19 Complementary filter approach: $x-y$ positions and z-error position
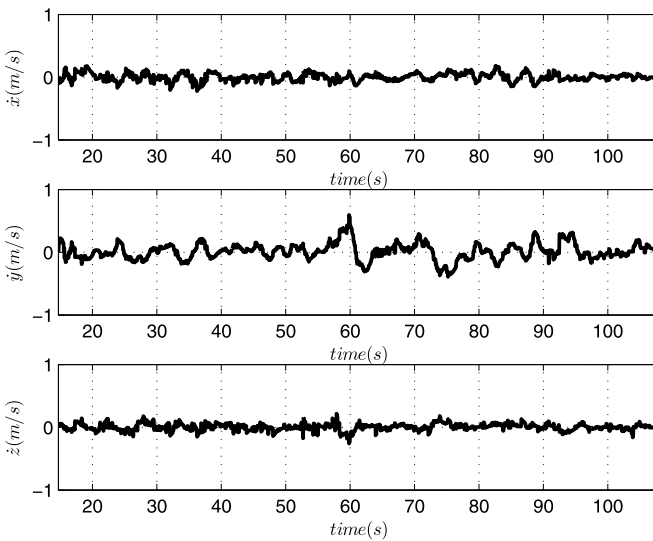
Fig. 7.20 Complementary filter approach: $x-y$ and z translational velocities

degraded after 50 seconds, it is considered that this can be caused by energy consumption issues.

The velocity computed by the Kalman filter, see Fig. 7.17, is the smoothest velocity signal obtained from the three state estimators. Figure 7.16 shows that the controller succeeds to maintain the x position of the helicopter around the origin,
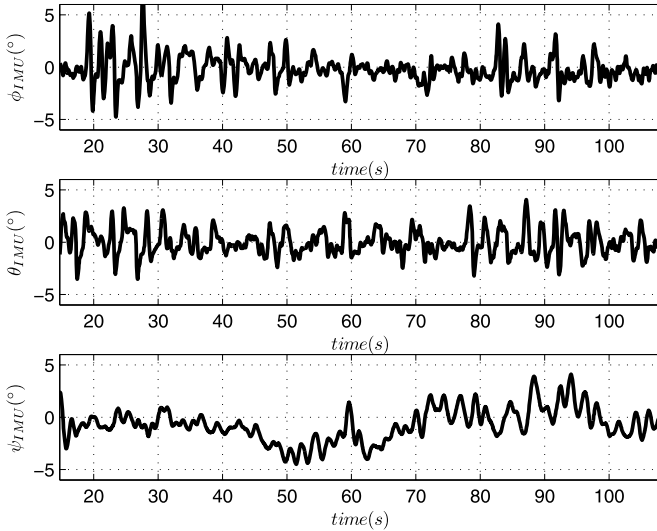
Fig. 7.21 Complementary filter approach: Behavior of the Euler angles

the y position around -0.5 m, and the altitude position with a maximum error of 0.4 m. However, the Euler angles of the platform while using this approach, see Fig. 7.18, as a result were more affected due to corrections in the position.

The controller performance when using the velocity estimated with the complementary filter (Fig. 7.20), represents a good balance between position stabilization and angular behavior. This velocity estimation is smoother than the Luenberger observer estimated velocity, however, it is evidently more attenuated than the Kalman filter estimated velocity. The position of the helicopter shown, see Fig. 7.19, was maintained, within an acceptable margin, for almost 100 seconds. This was the longest experiment achieved during our tests.

Tables 7.1 and 7.2 show the mean and standard deviation values for the position and Euler angles signals obtained during the experiments. Note that Tables 7.1 and 7.2 were computed with only one experiment for each state observer, considering the UAV in steady state. Note in these tables that the position obtained when using the Kalman filter has the lower standard deviation. Note also that when using the complementary filter the pitch and roll Euler angles are less disturbed while correcting the position. If the control objective concerns also the energy consumption, the complementary filter can be considered as a good option. An image of the quad-rotor while performing the tests can be seen in Fig. 7.22.

7.2.5 Final Comments

A Luenberger observer, a Kalman filter and a complementary filter were implemented and compared experimentally to estimate the translational velocity of a

Fig. 7.22 An image of the quad-rotor UAV while performing the experiments
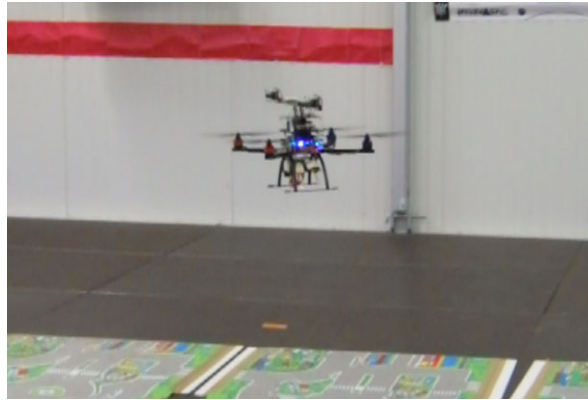
Table 7.1 Mean values of Euler angles and position

Parameter	Luenberger observer	Kalman filter	Complementary filter
Roll angle	−0.1866°	0.4729°	0.0831°
Pitch angle	−0.0031°	0.1382°	−0.1618°
Yaw angle	−0.7076°	−0.0210°	−1.2504°
X position	−0.0774 m	0.0021 m	−0.0527 m
Y position	−0.0423 m	−0.5433 m	−0.3802 m
e_Z error	0.4634 m	−0.2157 m	0.1248 m

Table 7.2 Standard deviation of Euler angles and position

Parameter	Luenberger observer	Kalman filter	Complementary filter
Roll angle	1.3030°	1.3604°	1.1058°
Pitch angle	1.2175°	1.4999°	0.8229°
Yaw angle	1.5601°	1.1205°	1.7270°
X position	0.0992 m	0.0575 m	0.1628 m
Y position	0.3016 m	0.2905 m	0.5746 m
e_Z error	0.1878 m	0.1248 m	0.1733 m

quad-rotor UAV equipped with an imaging and inertial sensing system. The stereo vision system was used to estimate the (x, y, z) 3-dimensional position of the aerial vehicle with respect to a fixed inertial frame, while an IMU was used to provide the linear accelerations experienced by the platform. A control strategy was implemented onboard to stabilize the position of the helicopter while using the estimated position and velocity data.

Experimental results have shown that the response of the system is smoother and that the vehicle position is closer to the desired values when using the velocity estimated by the Kalman filter.

7.3 Concluding Remarks

This chapter presented the development of a quad-rotor robotic platform equipped with a visual, inertial and altitude sensing system, which allows the helicopter to fully estimate its states without using GPS or artificial visual landmarks. The research objective consists of enabling the UAV to autonomously perform take-off, relative positioning, navigation and landing, when evolving in unstructured, indoors, and GPS-denied environments.

A stereo visual odometry algorithm was implemented with the purpose of estimating the vehicle ego-motion, in all six degrees of freedom, from the images provided by the cameras. The acceleration experienced by the vehicle was measured by an IMU, and, in addition, an ultrasonic sensor provided an additional estimation of the helicopter altitude. The output of these three sensors was combined in a simple Kalman filter strategy, allowing the estimation of the quad-rotor 3-dimensional position and translational velocity in an accurate way. The effectiveness of the proposed approach was evaluated in real-time experiments indoors, achieving an acceptable estimation of the quad-rotor's translational dynamics. The vehicle's states are provided at a rate of 13 Hz, which is an adequate frequency for performing vision-based position control tasks.

Once the desired functionality of the proposed system was achieved, a Luenberger observer, a Kalman filter and a complementary filter were implemented and compared in real-time experiments, with the purpose of identifying the most effective approach for combining visual odometry with inertial measurements. By using the estimated position and velocity data, a nested saturations control strategy was implemented onboard for stabilizing the vehicle when flying. It was experimentally found that the response of the system is smoother and that the vehicle's position is closer to the desired values when using the velocity estimated by the Kalman filter.

Chapter 8
Conclusions and Future Work

8.1 Conclusions

The research work presented in this document is devoted to the development of original and robust control methods aiming at performing autonomous hover flights and navigation of a quad-rotor robotic helicopter. In order to achieve a fully functional UAV platform, different research areas have been addressed, such as control, mechanics, computer vision and embedded systems. The research work conducted on those subjects provided several results, which are detailed next.

Development of a Well Suited Quad-Rotor Platform A set of three quad-rotor platforms were designed and evaluated in real-time experiments. Thanks to the experience obtained from these developments, each vehicle has an important improvement with respect to its predecessors. The efforts in this subject gave as result a quad-rotor robotic helicopter called the "Improved X-Flyer", well suited for performing take-off, hovering, navigation and landing tasks. This vehicle has proven being easy to control, robust with respect to external perturbations and with an adequate time of autonomy. In addition, it has the important characteristic of being able to lift a considerably amount of payload, which enables the vehicle to carry a considerable set of embedded electronics and sensors.

A supervisory ground station for the aerial vehicle was also presented. This complementary system allows flight supervision, as well as performing manually guided flights and vision-based control. Also, it is possible to recover all the information describing the vehicle's states, which helps to analyze the tests performed.

Control System for Improving the Attitude Stabilization An embedded control system for improving the attitude stabilization of a quad-rotor mini UAV was presented and tested in real-time applications. The control strategy uses low cost components and includes an extra control loop based on motor armature current feedback. It was shown that this methodology significantly improves the performance of the quad-rotor attitude stability, reducing pitch and roll errors with respect

to the horizontal plane. In addition, it has been observed experimentally that the controller is robust with respect to external disturbances. This enhancement enables the vehicle to perform autonomous tasks requiring a more precise attitude stabilization, like vision-based positioning and navigation.

Vision-Based Quad-Rotor Control Using a Monocular Imaging System   With the purpose of stabilizing the 3-dimensional position and translational velocity of the quad-rotor helicopter, two different vision-based strategies were presented. The complete system used in this application consists of a calibrated camera onboard the UAV, an artificial marker placed on ground, the imaging processing and control algorithms running on the supervisory ground station PC, and a wireless link between the helicopter and the supervisory ground station.

In the first methodology, an imaging algorithm consisting of an homography estimation technique, in combination with a feature detector and an optical flow computation, allowed estimating the 3-dimensional position and translational speed of the helicopter with respect to a landing pad on the ground. The experimental application was successfully performed indoors showing that the quad-rotor was stabilized at a selected position above the landing pad.

The second methodology presented consists of a computer vision algorithm which estimates the helicopter's altitude, lateral position and forward velocity, during flights over a road model. It has been shown that the visual information extracted allows the development of control strategies for performing hover flights and forward flight at constant speed.

It is also worth mentioning that both vision-based control strategies presented do not cause instability on the vehicle's attitude while correcting its position.

Study of Nonlinear Controllers Comparison   Determining which control strategy is better for stabilizing a quad-rotor motivated the performance of a comparison between three different control strategies: nested saturations, backstepping and sliding modes. For such tests, the quad-rotor translational dynamics were provided by the monocular imaging system.

Although all strategies tested achieve hovering flight, smoother translational and angular behavior is obtained when using the nested saturation controller. It ensures a smoother vehicle behavior and less angular corrections, which reduces the energy consumption with respect to the other two controllers.

Development of an Imaging, Inertial and Altitude Sensing System   The quad-rotor control strategy requires knowledge of the aircraft's linear and angular velocity and position. Evidently, if the quad-rotor evolves in unstructured, indoors GPS-denied environments, achieving autonomous tasks becomes very complicated.

To overcome such a situation, a quad-rotor UAV was equipped with an imaging, inertial and altitude sensing system, which enables the helicopter to fully estimate its states without using GPS or artificial visual landmarks. A Kalman filter strategy was implemented to combine visual odometry measurements, IMU acceleration data and altitude sensor signals, with the purpose of providing accurate estimations of the states describing the behavior of the platform.

It was shown that the proposed sensing system is capable of determining the quad-rotor 3-dimensional position and translational velocities in an accurate way. Such information is estimated at a rate of 13 Hz, which is adequate for real-time control purposes

Study of Different State Estimation Algorithms   The payload of miniature UAVs is severely restricted with the purpose of avoiding unnecessary energy consumption and increasing autonomy. For this reason, the number and the weight of onboard sensors is limited. This situation motivated the search of an efficient state observer for the quad-rotor.

A Luenberger observer, a Kalman filter and a complementary filter were implemented and compared experimentally to estimate the translational velocity of the quad-rotor equipped with stereo imaging and inertial sensors. The vision system was used to estimate the 3-dimensional position of the aerial vehicle, while the IMU provided its linear accelerations. A control strategy was implemented onboard to stabilize the position of the helicopter. Experimental results have shown that the response of the system is smoother and that the vehicle position is closer to the desired values when using the velocity estimated by the Kalman filter.

8.2 Future Work

After the design of the attitude sensory system, which has been successfully developed during the experimental results, the imaging sensing system is the most critical sensor for the UAV. Performing image processing in the remote ground station generates problems which render more difficult the execution of a more realistic mission. The range of image transmission is limited, therefore the UAV must fly close to the supervisory station to ensure image reception. Furthermore, the sampling rate for the vision-based control is relatively low. In order to overcome such problems, a version of the imaging and inertial algorithm that runs entirely in the Flying Netbook computer could be developed. This improvement will eliminate the wireless transmission delays of the required data to the onboard autopilot. The Flying Netbook embedded program presented provides the helicopter states at 7 Hz, which is unfortunately not enough for controlling the position of the vehicle in real-time application. Modifications and improvements must be conducted in order to speed up the estimation frequency rate, aiming at a minimum working frequency of 15 Hz. It could be developed also the interconnection stage (hardware and software) between the Flying Netbook computer (or other microprocessor) and the autopilot. Another issue that must be tackled is the improvement of sensing techniques in outdoors environments. The final objective of this future work is to achieve a complete image processing and vision-based positioning control onboard the quad-rotor.

A further improvement to the robotic platform can consist of the implementation of a vision-based system capable of performing at night or under poor illumination conditions. Aiming at this goal, an infrared camera can be used since this kind of device can effectively work at night or under a weak illuminated condition. However,

the infrared camera's FOV is usually narrower when compared to a CCD camera's FOV. For this reason, the requirements of the mission to be performed must be evaluated before deciding to install a normal or infrared camera.

The experience obtained from the development of the quad-rotor experimental platform pointed out that tuning the controller is a complicated task when implementing real-time systems. Since the quad-rotor's parameters are usually not known, the controller gains are adjusted empirically by performing several trials. The development of an algorithm for system identification, allowing the estimation of the vehicle's parameters, seems to be a very pertinent subject. Addressing this issue will help reduce the time devoted to tuning the controllers.

References

1. Achtelik, M.: Vision-based pose estimation for autonomous micro aerial vehicles in GPS-denied areas. Master's Thesis. Technische Universitat Munchen, April 2009
2. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In: Unmanned Systems Technology XI, April 2009. Proc. of SPIE, vol. 7332 (2009)
3. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In: First Symposium on Indoor Flight Issues, Mayagüez, Puerto Rico, July 2009
4. Ahrens, S., Levine, D., Andrews, G., How, J.P.: Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments. In: IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009, pp. 2643–2648 (2009)
5. Alderete, T.S.: Simulator aero model implementation. NASA Ames Research Center, Moffett Field, California (1995)
6. Altuğ, E., Ostrowski, J.P., Taylor, C.J.: Control of a quadrotor helicopter using dual camera visual feedback. Int. J. Robot. Res. **24**(5), 329–341 (2005)
7. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. Int. J. Comput. Vis. **2**, 283–310 (1989)
8. Ascending Technologies. http://www.asctec.de/, June 2011
9. Austin, R.: Unmanned Aircraft Systems: UAV Design, Development and Deployment. Wiley, New York (2010)
10. Barrows, G.L., Neely, C.: Mixed-mode VLSI optic flow sensor for in-flight control of a micro air vehicle. In: Critical Technologies for the Future of Computing, vol. 4109, pp. 52–63. SPIE, Bellingham (2000)
11. Benzemrane, K., Damm, G., Santosuosso, G.L.: Adaptive observer and Kalman filtering. In: International Federation of Automatic Control, Seoul, Korea, July 2008
12. Benzemrane, K., Santosuosso, G.L., Damm, G.: Unmanned aerial vehicle speed estimation via nonlinear adaptive observers. In: American Control Conference, New York, USA, July 2007
13. Blosch, M., Weiss, S., Scaramuzza, D., Siegwart, R.: Vision based MAV navigation in unknown and unstructured environments. In: IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA, May 2010, pp. 21–28 (2010)
14. Bouabdallah, S., Noth, A.: PID vs LQ control techniques applied to an indoor micro quadrotor. In: IEEE International Conference on Intelligent Robots and Systems, Sendai, Japan, October 2004
15. Bouabdallah, S., Siegwart, R.: Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In: IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005

16. Bouguet, J.Y.: Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, Intel Corporation, Microprocessor Research Labs, Technical report (1999)
17. Bouguet, J.Y.: Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html (2011)
18. Bouguet, J.Y.: Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm. Microprocessor Research Labs, Intel Corporation (2002)
19. Boutayeb, M., Richard, E., Rafaralahy, H., Ali, H.S., Zaloylo, G.: A simple time-varying observer for speed estimation of UAV. In: International Federation of Automatic Control, Seoul, Korea, July 2008
20. Bradski, G., Kaehler, A.: Learning OpenCV. O'Reilly Media, Sebastopol (2008)
21. Brown, R.G., Hwang, P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering, 3rd edn. Wiley, New York (1996)
22. Buskey, G.D., Roberts, J., Corke, P.I., Wyeth, G.F.: Helicopter automation using a low-cost sensing system. Comput. Control Eng. J. **15**(2), 8–9 (2004)
23. Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Vision-based odometry and slam for medium and high altitude flying UAVs. J. Intell. Robot. Syst. **54**, 137–161 (2009)
24. Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Vision-based odometry and slam for medium and high altitude flying UAVs. J. Intell. Robot. Syst. **54**(1–3), 137–161 (2009)
25. Campoy, P., Garcia, J., Barrientos, A., Cerro, J., Aguirre, I., Roa, A., Garcia, R., Munoz, J.M.: An autonomous helicopter guided by computer vision for overhead power cable inspection. In: 5th International Conference in Live Maintenance, Madrid, Spain (2000)
26. Campoy, P., Correa, J.F., Mondragon, I., Martinez, C., Olivare, M., Mejias, L., Artieda, J.: Computer vision onboard UAV for civilian tasks. J. Intell. Robot. Syst. (2008)
27. Castillo, P., Dzul, A., Lozano, R.: Real time stabilization and tracking of a four-rotor mini rotorcraft. IEEE Trans. Control Syst. Technol. **12**(4), 510–515 (2004)
28. Chen, C.-T.: Linear Systems Theory and Design. Oxford University Press, London (1998)
29. Cheng, Y., Maimone, M.W., Matthies, L.: Visual odometry on the Mars exploration rovers. IEEE Robot. Autom. Mag. **13**(2), 54–62 (2006)
30. Corke, P.: An inertial and visual sensing system for a small autonomous helicopter. J. Robot. Syst. **21**(2), 43–51 (2004)
31. Corke, P.I., Strelow, D., Singh, S.: Omnidirectional visual odometry for a planetary rover. In: IEEE International Conference on Intelligent Robots and Systems. Sendai, Japan, September 2004
32. Demonceaux, C., Vasseur, P., Pégard, C.: Omnidirectional vision on UAV for attitude computation. In: IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, May 2006
33. Etkin, B., Reid, L.D.: Dynamics of Flight. Wiley, New York (1959)
34. Eynard, D., Vasseur, P., Demonceaux, C., Fremont, V.: UAV altitude estimation by mixed stereoscopic vision. In: IEEE Int. Conf. on Intelligent Robots and Systems, IROS'10, Taipei, Taiwan, October 2010
35. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
36. Fleet, D.J., Jepson, A.D.: Computation of component image velocity from local phase information. Int. J. Comput. Vis. **5**, 77–104 (1990)
37. García Carrillo, L.R., Rondon, E., Sanchez, A., Dzul, A., Lozano, R.: Stabilization and trajectory tracking of a quad rotor using vision. J. Intell. Robot. Syst. **61**(1–4), 103–118 (2011)
38. Goldstein, H.: Classical Mechanics. Addison Wesley, Reading (1980)
39. Guenard, N., Hamel, T., Mahony, R.: A practical visual servo control for a unmanned aerial vehicle. IEEE Trans. Robot. **24**(2), 331–341 (2008)
40. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
41. He, R., Prentice, S., Roy, N.: Planning in information space for a quadrotor helicopter in a GPS-denied environment. In: IEEE International Conference on Robotics and Automation,

Pasadena, CA, USA, May 2008

42. Heeger, D.J.: Optical flow using spatio temporal filters. Int. J. Comput. Vis. **1**, 279–302 (1988)

43. Hespanha, J., Yakimenko, O., Kaminer, I., Pascoal, A.: Linear parametrically varying systems with brief instabilities: An application to integrated vision/imu navigation. IEEE Trans. Aerosp. Electron. Syst. **40**(3), 889–902 (2004)

44. Horaud, R., Conio, B., Leboulleux, O., Lacolle, B.: An analytic solution for the perspective 4-point problem. Comput. Vis. Graph. Image Process. **47**, 33–44 (1989)

45. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**, 185–204 (1981)

46. Hrabar, S., Sukhatme, G.: Vision-based navigation through urban canyons. J. Field Robot. **26**(5), 431–452 (2009)

47. Hrabar, S., Sukhatme, G.S.: Omnidirectional vision for an autonomous helicopter. In: IEEE International Conference on Robotics and Automation, Taiwan, May 2003, pp. 558–563 (2003)

48. IDS Imaging Development Systems GmbH. http://www.ids-imaging.com/, June 2011

49. Jarrell, J., Gu, Y., Seanor, B., Napolitano, M.: Aircraft attitude, position, and velocity determination using sensor fusion. In: AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, USA (2008)

50. Johnson, A., Montgomery, S., Matthies, L.: Vision guided landing of an autonomous helicopter in hazardous terrain. In: International Conference on Robotics and Automation, Barcelona, Spain, pp. 4470–4475 (2005)

51. Kelly, J., Saripalli, S., Sukhatme, G.S.: Combined visual and inertial navigation for an unmanned aerial vehicle. In: International Conference on Field and Service Robotics, Chamonix, France, July 2007

52. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice-Hall, San Francisco (2002)

53. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of Electric Machinery and Drive Systems. IEEE Press Series on Power Engineering, vol. 10 (2008)

54. Krishnan, R.: Electric Motor Drives: Modeling, Analysis, and Control. Prentice Hall, New York (2001)

55. Levin, A., Szeliski, R.: Visual odometry and map correlation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, June 2004

56. Lim, K.B., Shin, J.-Y., Cooper, E.G., Moerder, D.D., Khong, T.H., Smith, M.F.: An overview of the NASA flying test platform research. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, Texas, USA (2003)

57. Lozano, R., Salazar Cruz, S., Sanchez, A., Fantoni, I.: Discrete-time stabilization of integrators in cascade with bounded inputs and delay. In: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, July 2006

58. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. DARPA IU Workshop, pp. 121–130 (1981)

59. Ludington, B., Johnson, E.N., Vachtsevanos, G.J.: Vision based navigation and target tracking for unmanned aerial vehicles. In: Advances in Unmanned Aerial Vehicles, pp. 245–266. Springer, Berlin (2007)

60. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3-D Vision: From Images to Geometric Models. Springer, Berlin (2003)

61. MikroKopter. http://www.mikrokopter.de/ucwiki/en/mikrokopter, June 2011

62. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, July 2004, pp. 652–659 (2004)

63. Ogata, K.: Discrete-Time Control Systems. Prentice-Hall, New York (1995)

64. Open Source Computer Vision. http://opencv.willowgarage.com (2011)

65. Proctor, A.A., Johnson, E.N., Apker, T.B.: Vision-only control and guidance for aircraft. J. Field Robot. **23**(10), 863–890 (2006)

66. Romero, H., Benosman, R., Lozano, R.: Stabilization and location of a four rotor helicopter applying vision. In: Proceedings of the American Control Conference, Minneapolis, MN, USA, June 2006, pp. 3930–3936 (2006)

67. Romero, H., Salazar, S., Lozano, R.: Stabilization of an eight-rotor UAV using optical flow. IEEE Trans. Robot. **25**(4), 809–817 (2009)
68. Rondon, E., Fantoni-Coichot, I., Sanchez, A., Sanahuja, G.: Optical flow-based controller for reactive and relative navigation dedicated to a four rotor rotorcraft. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, USA, October 2009, pp. 684–689 (2009)
69. Rothhaar, P.M., Moerder, D.D., Lim, K.B.: Hovering dual-spin vehicle groundwork for bias momentum sizing validation experiment. In: AIAA Atmospheric Flight Mechanics Conference and Exhibit, Honolulu, Hawaii, USA (2008)
70. Saberi, A., Kokotovic, P.V., Sussmann, H.J.: Global stabilization of partially linear composite systems. SIAM J. Control Optim. **28**(6), 1491–1503 (1990)
71. Sanchez, A., Garcia Carrillo, L.R., Rondon, E., Lozano, R., Garcia, O.: Hovering flight improvement of a quad-rotor mini UAV using brushless DC motors. J. Intell. Robot. Syst. **61**(1–4), 85–101 (2011)
72. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Visually guided landing of an unmanned aerial vehicle. IEEE Trans. Robot. Autom. **19**(3), 371–381 (2003)
73. Saripalli, S., Sukhatme, G., Mejias, L.O., Campoy Cervera, P.: Detection and tracing of external features in an urban environment using an autonomous helicopter. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp. 3972–3977 (2005)
74. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA, pp. 593–600 (1994)
75. Singh, A.: An estimation-theoretic framework for image-flow computation. In: Proc. IEEE ICCV, Osaka, pp. 168–177 (1990)
76. Sünderhauf, N., Protzel, P.: Stereo odometry—a review of approaches. Technical Report, Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, March 2007
77. Teel, A.R.: Global stabilization and restricted tracking for multiple integrators with bounded controls. In: Systems and Control Letters, vol. 18, pp. 165–171 (1992)
78. Texas Instruments. TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812. Digital Signal Processors. Data Manual (2001)
79. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 376–380 (1991)
80. Waxman, A.M., Ullman, S.: Surface structure and three dimensional motion from image flow kinematics. Int. J. Robot. Res. **4**(3), 72–94 (1985)
81. Wertz, J.R.: Spacecraft Attitude Determination and Control. Kluwer Academic, Dordrecht (1978)
82. Yu, Z., Nonami, K., Shin, J., Celestino, D.: 3D vision based landing control of a small scale autonomous helicopter. Int. J. Adv. Robot. Syst. **4**(1), 51–56 (2007)
83. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, pp. 666–673 (1999)
84. Zufferey, J.C., Floreano, D.: Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. In: IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005

Index