Vision based navigation for micro helicopters

Author(s):
Weiss, Stephan M.

Publication Date:
2012

Permanent Link:

https://doi.org/10.3929/ethz-a-007344020 →

DISS. ETH NO. 20305

Vision Based Navigation for Micro Helicopters

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

Stephan M. Weiss

Master of Science in Electrical Engineering and Information Technology
Eidgenössische Technische Hochschule (ETH) Zurich, Switzerland

Born July 16th, 1981 in Caracas, Venezuela
Citizen of Switzerland and Venezuela

accepted on the recommendation of

Prof. Roland Siegwart,	ETH Zurich
Prof. Vijay Kumar,	University of Pennsylvania (UPENN)
Dr. Agostino Martinelli,	Institut de Recheche en Informatique et Automatique (INRIA)

2012

to my parents

Acknowledgments

First of all, I would like to thank Roland Siegwart for having offered me the possibility of doing a PhD under his supervision. He provided me a highly inspiring environment and all the necessary resources to achieve the present work. Being part of Roland Siegwart's group – the Autonomous Systems Lab (ASL) – was a highly appreciated experience and led to many happy moments.

It was mainly Agostino Martinelli who taught me the knowledge needed to fulfill the work as it is now. I am very happy about the coincidence that we were both on the same project (sFly) and that we could use this time to have many fruitful and – for me – educative discussions. Thus, it is a special honor to have Agostino in my committee. I am both proud and grateful that Vijay Kumar (GRASP, UPENN), worldwide renown expert and leader in mobile robotics, agreed to complete my committee.

Nowadays, a (scientific) work is rarely accomplished by one single person – neither is this work. In fact, without the support of many people, this work would not have been possible. This is, in particular, my office colleague Markus Achtelik who invested countless hours to provide a working platform for testing of the here presented algorithms. Not only provided he a working platform, but also was more than once an excellent safety-pilot preventing crashes of the MAV when my algorithms failed. Also, I am happy to have shared the office with Laurent Kneip who gave highly appreciated constructive feedback to my work in order to further improve it. I also received honest and constructive feedback from my second post-doc supervisor Margarita Chli.

I am thankful that Davide Scaramuzza as my first post-doc supervisor guided me well through the beginning of my time as a PhD candidate. Also during this time, I supervised with Davide three Bachelor students who essentially laid the foundation of this dissertation. I would like to express my deepest gratitude to Michael Blösch, Daniel Eberli and Gabriel Nützi who achieved a vision based autonomous MAV navigation framework which was

the first of its kind in the research community and was a solid proof of concept for this dissertation.

During my time as a PhD candidate I had the chance to do a research visit at Daniela Rus' Lab (DRL) at MIT. I want to thank Daniela very much for having made this visit possible. In fact, it was only thanks to the huge effort of Daniela and Andreas Breitenmoser that I was able to do this unforgettable visit. Thanks to this visit, I got the chance to meet Jonathan Kelly in person and to have very fruitful discussions in the field of this dissertation. I thank Jonathan not only for these discussions at MIT but also for his prompt and detailed replies by email previously to my visit to all may countless questions about his work on observability of non-linear systems.

This work would not have been possible without the support on hardware and infrastructure. My thanks go to Thomas Baumgartner, Stefan Bertschi, Markus Bühler and Dario Fenner for a fabulous support during this time. Similarly, I like to thank Luciana Borsatti, Cornelia Della Casa, Margot Fox, and Eve Lassere for their support in administrative questions.

The inspiration at ASL came not only thanks to the high quality researcher and good infrastructure, but also thanks to the friendly and even fun atmosphere – also (and maybe in particular?) during conferences abroad. I wish to thank all my colleagues for making this time at ASL unforgettable.

And finally, I am deeply grateful to my family for all the support they gave me to reach this point and to my closest friends who supported me in every difficult and fun situation.

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n. 231855 (sFly).

Zurich, January 2012

Abstract

In this dissertation, we study the issues of vehicle state estimation and sensor self-calibration which arise while navigating a micro helicopter in large and initially unknown environments. While proper vehicle state estimation allows *long-term* navigation, sensor self-calibration renders the vehicle a *power-on-and-go* system without the need of prior and offline calibration steps. Due to the inherent payload limitation of airborne vehicles like, and especially, a micro helicopter and the restricted availability of global localization information in indoor or urban environments, we focus on *vision based* methods in order to achieve our goals. Despite this focus on processing visual cues, we analyze a general and modular approach which allows the use of a variety of different sensor types.

In vision based methods, the camera needs special attention. In contrast to other sensors, vision sensors typically yield vast information that needs complex strategies to permit use in real-time and on computationally constrained platforms. We show that our map-based visual odometry strategy derived from a state-of-the-art structure-from-motion framework is particularly suitable for locally stable, pose controlled flight. Issues concerning drifts and robustness are analyzed and discussed with respect to the original framework. A map-less strategy based on optical flow and inertial constraints is presented in order to mitigate other map-related issues such as map losses and initialization procedures. While in the map-based approach the camera acts as a real-time pose sensor, in the map-less approach it acts as a real-time body-velocity sensor capable of velocity-control the airborne vehicle. We show that both approaches can be combined in a unifying and complementary framework to mitigate each others weaknesses. In all cases, the algorithms are capable of running in real-time on a computationally constrained platform.

For aerial vehicle navigation we study a statistical and modular sensor-fusion strategy. We focus on the optimal state estimation of the vehicle to avoid additional, computationally heavy control approaches. In particular, we discuss the metric recovery of an arbitrarily scaled pose or body-velocity

measured by the camera as well as the recovery of different drifts the vision based strategies suffer from. Our modular approach allows to treat the camera as a black box sensor. This renders the computational complexity of our sensor-fusion strategy constant and applicable to large environments and long-term navigation. The modularity of our system also allows the addition of a variety of different sensor types. Based on a concise non-linear observability analysis using differential geometry we isolate the drift states and unobservable modes of a system with a given sensor setup and provide extensive insight to the observable modes. We discuss the observability of both, the on-line state estimation of the vehicle, and self-calibration of the system. The latter includes all inter-sensor calibration states including the intrinsic inertial characteristics. For the unobservable modes, we highlight their dimensions in the state space by analyzing the continuous symmetries and indistinguishable regions of the system. Based on this, we present the effects of combining a variety of sensors in order to not only estimate the vehicle state but also to recover different drift and (inter-)sensor calibration states of the system.

The theoretical analysis of the non-linear and time continuous multi-sensor system is necessary, but does not provide a sufficient condition for the observability of the implemented linearized and time discrete system. Thus, we analyze the convergence behavior in extensive simulations. Our approach is further verified and tested by implementing it on real hardware and testing it on real data. We perform these tests indoor in a controlled way while having accurate ground truth for verification. Furthermore we perform tests outdoor in a large environment. These outdoor tests prove the validity of the presented real-time and on-board vision algorithms converting the camera into black boxed sensor. Also, they show the capability of the entire sensor-fusion framework to convert an aerial vehicle into a power-on-and-go system for real-world, large-scale and long-term operations. The insights gained from the theoretical analysis are used to handle sensor drifts and sensor dropouts up to switching between completely different sensor suites in flight as it is required during indoor-outdoor transitions. This summarizes the dissertation as a thorough work going from initial sensor preparation to detailed theory through careful implementation and down to successful real-world testing for tomorrows autonomous aerial vehicles.

Keywords: Multi-Sensor Fusion, Sensor Self-Calibration, Visual-Inertial Navigation, Non-Linear Observability, Continuous Symmetries, Power-on-and-go-Systems

Kurzfassung

In dieser Dissertation untersuchen wir die Fragen zur Zustandsschätzung und Sensor-Selbstkalibrierung, welche bei der Navigation von Mikro-Helikopter in grossen und zunächst unbekannten Umgebungen entstehen. Während eine präzise Zustandsschätzung eine *langfristige* Navigation ermöglicht, macht die Sensor-Selbstkalibrierung aus dem Roboter ein *power-on-and-go* System ohne der Notwendigkeit von Vor- und Offline-Kalibrierungs-Schritten. Aufgrund der inhärenten Begrenzung von Nutzlast bei Mikro-Helikopter und der eingeschränkten Verfügbarkeit von globalen Informationen in Innen- oder Stadt-Umgebungen, konzentrieren wir uns auf *visuell* basierte Methoden, um unsere Ziele zu erreichen. Trotz dieser Fokussierung auf die Verarbeitung visueller Stimuli analysieren wir einen allgemeinen und modularen Ansatz, welcher die Verwendung einer Vielzahl von verschiedene Sensortypen ermöglicht.

Im Gegensatz zu anderen Sensoren liefern Kamera typischerweise exzessive Informationen die komplexe Strategien benötigen, um den Einsatz in Echtzeit und auf rechnerisch eingeschränkt Plattformen zu erlauben. Wir zeigen, dass sich unsere Karten-basierte Strategie mit visueller Odometrie besonders eignet für einen lokal stabilen, kontrollierten Flug. Fragen im Zusammenhang mit Drifts und Robustheit werden analysiert und diskutiert. Eine Karten-lose Strategie, basierend auf optischem Fluss und inertiale Einschränkungen, wird präsentiert, um Probleme wie Karten-Verlust und Initialisierung bei Karten-basierten Ansätzen zu mildern. Während der Karten-basierte Ansatz die Kamera als Echtzeit-Positions-Sensor darstellt, kann der Karten-lose Ansatz als ein Echtzeit-Körpergeschwindigkeitssensor gesehen werden, welcher den Roboter in Geschwindigkeit steuern kann. Wir zeigen, dass sich beide Ansätze in einem einzigen und sich ergänzenden Rahmen kombinieren lassen, um so die Schwächen des jeweilig anderen Ansatzes zu mildern.

Für die Helikopter-Navigation studieren wir eine statistische, modulare Sensor-Fusionierungs-Strategie. Insbesondere diskutieren wir die metrische

Schätzung einer willkürlich skalierten Position oder Körpergeschwindigkeit von der Messung der Kamera, sowie die Schätzung von verschiedenen Drifts der visuellen Algorithmen. Unser modulares Konzept erlaubt es, die Kamera als geschlossene Box zu behandeln. Dies macht die Komplexität des Rechenaufwandes unserer Sensor-Fusionierungs-Strategie konstant und deshalb brauchbar für grosse Umgebungen und langfristige Navigation. Die Modularität unseres Systems ermöglicht auch die Zugabe einer Vielzahl von verschiedenen Sensortypen. Basierend auf einer ausführlichen nichtlinearen Beobachtbarkeitsanalyse mittels Differentialgeometrie isolieren wir die Drift-Zustände und die beobachtbaren Modi eines Systems und bieten umfassenden Einblick in die nicht-beobachtbaren Modi. Wir diskutieren die Beobachtbarkeit sowohl für die Online-Zustands-Schätzung des Helikopters, als auch für die Selbstkalibrierung des Systems. Letztere umfasst alle zwischen-Sensor Kalibrierungs-Zustände, darunter die intrinsischen Eigenschaften des Inertial-Sensors. Bei den nicht-beobachtbaren Modi beleuchten wir ihre Dimensionen im Zustandsraum durch Analyse der kontinuierlichen Symmetrien und ununterscheidbarer Bereiche des Systems.

Die theoretische Analyse des nicht-linearen und zeitkontinuierlichen Multisensorsystem ist notwendig, aber keineswegs eine hinreichende Bedingung für die Beobachtbarkeit des implementierten linearisierten und zeitdiskreten Systems. So analysieren wir das Konvergenzverhalten in umfangreichen Simulationen. Unser Ansatz wird weiter überprüft und getestet durch die Umsetzung auf echter Hardware und Tests auf realen Daten. Wir führen diese Tests innen in einer kontrollierten Art und Weise durch, während genaue Daten als Verifikation dienen. Darüber hinaus führen wir Tests im Freien in einer grossen Umgebung durch. Diese Tests beweisen die Wirksamkeit der vorgestellten Echtzeit- und On-Board-Algorithmen. Auch zeigen sie die Fähigkeit des gesamten Sensor-Fusionierungs-Systems, ein Helikopter in ein Power-On-and-Go-System für lange und echte Einsätze zu verwandeln. Die Erkenntnisse aus der theoretischen Analyse werden benutzt, um Sensor-Drifts und Sensor-Ausfälle bis zu Wechsel zwischen völlig unterschiedlichen Sensor-Modulen im Flug effizient zu handhaben. Dies fasst die Dissertation als gründliche Arbeit von der ersten Sensor-Vorbereitung, zur detaillierten theoretischen Analyse über eine sorgfältige Umsetzung bis zum erfolgreichen echten Testen der Fluggeräte der Zukunft zusammen.

Schlüsselworte: Multi-Sensor, Sensor-Fusionierung, Selbst-Kalibration, Visuell-Inertiale Navigation, Nichtlineare Beobachtbarkeit, Kontinuierlichen Symmetrien, Power-on-and-Go-Systeme

Acronyms

ASL	Autonomous Systems Laboratory
DoF	degrees of freedom
EKF	extended Kalman filter
ETH	Eidgenössische Technische Hochschule
FCU	flight control unit
HLP	high level processor
IMU	inertial measurement unit
KF	Kalman filter
LLP	low level processor
LQG	linear quadratic gain
LTR	loop transfer recovery
MAV	micro aerial vehicle
OF	optical flow
SLAM	simultaneous localization and mapping
UAV	unmanned aerial vehicle
UKF	unscented Kalman filter
VSLAM	visual simultaneous localization and mapping

Contents

1 Introduction **1**
- 1.1 Motivation and Objectives . . . 6
- 1.2 Current State of Research . . . 8
- 1.3 Contributions . . . 17
  - 1.3.1 Camera as a Motion Sensor . . . 18
  - 1.3.2 State Estimation and Sensor Self-Calibration . . . 19
- 1.4 Structure of this Dissertation . . . 21

2 Camera as a Motion Sensor **23**
- 2.1 Map Based Approach: The Camera as a Pose Sensor . . . 24
  - 2.1.1 Description of the Visual SLAM Algorithm . . . 25
  - 2.1.2 Adaptations: From Visual SLAM to Onboard Visual Odometry . . . 28
  - 2.1.3 Feasibility Study . . . 43
- 2.2 Map Less Approach: The Camera as a Body-Velocity Sensor 47
  - 2.2.1 2D Continuous "8-Point" Algorithm . . . 48
  - 2.2.2 Performance of the Inertial-Optical Flow Framework . 53
- 2.3 Conclusion . . . 58
  - 2.3.1 Camera as a 6DoF Pose Sensor . . . 58
  - 2.3.2 Camera as a 3DoF Body-Velocity Sensor . . . 59

3 Modular Sensor Fusion: State Estimation and Sensor Self-Calibration **61**
- 3.1 Visual-Inertial MAV Navigation . . . 63
  - 3.1.1 Extended Kalman Filter Framework . . . 64
  - 3.1.2 False Pose Estimate Detection . . . 76
- 3.2 Modular Multisensor Systems . . . 78
- 3.3 Observability Analysis and Discussion . . . 79
  - 3.3.1 Non-Linear Observability Analysis . . . 80

3.3.2 Discussion of the Unobservable States 83
3.4 Additional Sensors for Drift Elimination 87
3.4.1 1DoF Drift Free Vision Box: Visual Compass 87
3.4.2 Fix World Direction: Three Axis Magnetometer . . . 87
3.4.3 Absolute Position Measurement: GPS 93
3.4.4 Combining Absolute Position and Fix World Direction 97
3.4.5 From Visual Pose to Visual Body-Velocity Sensor . . . 98
3.4.6 Multi-Sensor System Summary 102
3.4.7 Note on the Quality of Observability 113
3.4.8 Note on the Linearized and Discretized Systems . . . 116
3.5 Conclusion 117

4 Results: Performance Evaluation **121**
4.1 Simulation Results 122
4.1.1 Simulations on the Camera-IMU Translation 124
4.1.2 Simulations on the Camera-IMU Rotation 126
4.1.3 Simulations on the Visual Scale 127
4.1.4 Simulations on the IMU biases 131
4.1.5 Simulations on the Visual Attitude Drift 133
4.1.6 Simulations on the Additional States 133
4.2 Delay Compensated Distributed Implementation 135
4.2.1 System Setup 136
4.2.2 Distribution of Processing Tasks 137
4.2.3 Handling Measurement Delays 139
4.2.4 Handling Measurement Updates in Position Control . 141
4.2.5 Modular Design 141
4.2.6 Results on the Real Platform 141
4.2.7 Detecting and Handling Map Failure Modes 147
4.3 Results in a Large Environment 151
4.3.1 Vision Based Navigation 152
4.3.2 Multi-Sensor Navigation 160
4.4 Conclusion 168

5 Discussion and Conclusion **171**
5.1 Summary of this Dissertation 171
5.2 Discussion of Contributions 171
5.2.1 Camera as a Motion Sensor 172
5.2.2 Multi-Sensor Power-on-and-go-Systems 173
5.3 Applications and Research Outlook 177
5.3.1 Area Mapping and Coverage 178
5.3.2 Research Outlook 178

A Quantitative Results on the Real Platform **187**
A.1 Hovering 187
A.2 Dynamic Flight 188

Bibliography **191**

Curriculum Vitae **203**

Chapter 1

Introduction

The concept of combining different sources of information in order to improve the overall outcome is omnipresent in literally every scientific and non-scientific domain. Already, when we pose a question to someone, we weight the answer with our a-priori expected outcome to determine the level of trust we put in the additionally gained information. Or, it is common practice to read different newspapers about an incident in order to obtain a more objective view about a situation. Humans, and animals in general, not only perform this combination of different information sources consciously but also subconsciously. In fact, the subconscious combination of information occurs continuously by processing all outputs of our vast number of sensors. Apart from the traditionally taught senses — hearing, seeing, taste, touch, and smell — animals also have a sense of inclination and rotation, of the position of their joints with respect to each other and much more.

In nature, the sensor providing the richest external information is probably the eye. Appeared about 540 million years ago in fossils found from the lower Cambrian period (Parker (2011)) the eye evolved towards various different shapes and functions. Ranging nowadays from the “pinhole-model“ in animals like the Nautilus to a sophisticated ”lens-mounted color camera” in humans. Fig. 1.1 shows a rough overview on eye evolution. Although we feel the perception of a scene as an immediate action, it is evident, that we need some computing power to process the vast information of this sensor. This is reflected in the fact, that we start having difficulties to distinguish events of a temporal distance of less than 60ms (Breitmeyer and Ganz (1977)). Also, we developed a strategy to decide earlyon which images we should fully process. This is manifested in the fact, that, we normally do not process any information while we move the eyes. In contrast, if the scene is only illu-

minated during the eye movement and completely dark before and after the movement, we do process a blurred image as it is the only source of information (Campbell and Wurtz (1978)). In both cases we perceive the image, but extracting information is the costly process we subconsciously decide to do or not to do.

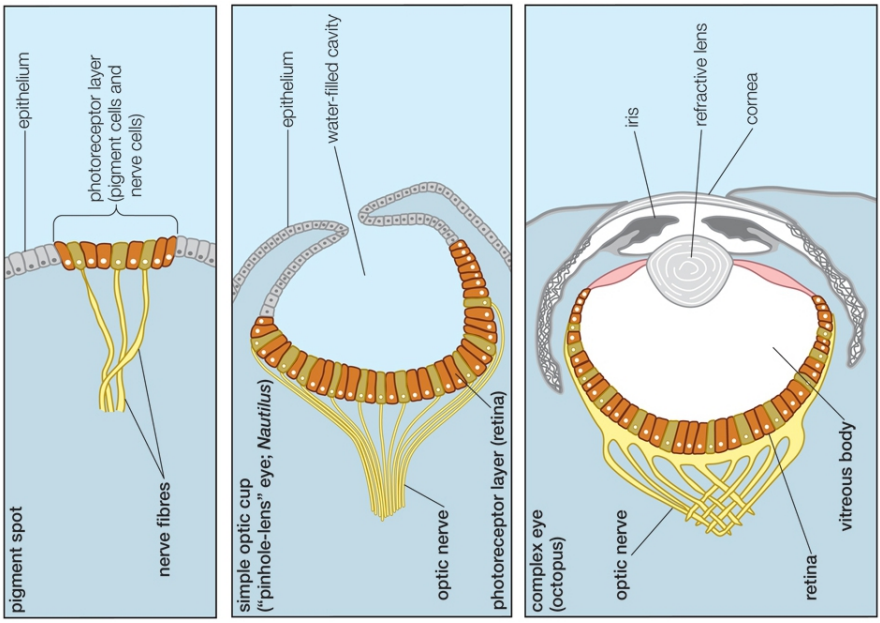


Figure 1.1: A rough overview on the evolution of the eye. Nowadays we still find all different stages of the eye in nature. The most comparable versions to cameras may range from the pinhole model of the Nautilus to the modern lens mounted camera model of the human. (picture modified from Enciclopaedia Britannica, Inc)

While this subconscious behavior evolved naturally to excellence in humans and animals, it has to be implemented explicitly in artificial devices. Image processing has long been considered in off-line tasks on only a very few, low-resolution images. It has been the expensive devices, the limited calculation power and the early algorithms, that prevented the extensive use of image processing in robotics. Instead, sonars, range finders and bulky laser devices found their applications in both industrial and mobile robots. These sensors gave mobile robots information about their surroundings leading to the emergence of the first applications of the Simultaneous Localization And Mapping (SLAM) problem. The approach worked well in artificial, inherently

bounded environments adapted to the sensor limitations. It is without doubt, however, that mobile robot applications need to be functional in larger and in particular natural environments. This requires more powerful sensing and more intelligent algorithms on the robotic platforms.

The mass production and miniaturization of cheap, yet qualitatively good cameras, initialized the price drop of these sensors and thus boosted research towards efficient algorithms for processing their vast data. It was mainly with the introduction of distinctive key-points (Bay et al. (2008); Harris and Stephens (1988); Lowe (2004); Rosten and Drummond (2006); Shi and Tomasi (1994)) that images could be processed rapidly enough to be used for robot navigation. In particular, the camera could then be seen as a sensor yielding either body-velocity (Zufferey and Floreano (2005)), incremental positions (Civera et al. (2010)) or even absolute position (Davison et al. (2007); Klein and Murray (2009)). While the cameras shrank sufficiently in size to be mounted on micro vehicles with a very limited payload budget, the algorithms still suffered from the need of large computation power and from accuracy deficiencies when used in long-term navigation. Using these algorithms as a basis to render the camera a visual pose sensor for low computation power platforms in large and natural environments is thus one of the declared goals in this dissertation.

Vision is certainly not the only sense nature is using for successful navigation and localization. Some animals, bats for example, developed sonar-like sensors with remarkable properties (Simmons and Stein (1980)). Others, like some birds, developed a sensitivity to the magnetic field of the Earth and even to celestial sources of directional information. It is in fact most interesting to know, that these birds perceive the inclination of the magnetic field. This leads to the problem that their sensor yields bimodal information at the magnetic equators. Technically speaking, in these areas, they enter an unobservable mode. Nature identified this unobservable mode and it was shown that birds can merge information of the magnetic field and celestial direction sources in order to circumvent these unobservable modes (Wiltschko and Wiltschko (1996)). Such remarkable sensor combinations can be found in a variety of examples. Most accessible for our understanding are daily situations when we merge visual cues with vestibular cues such as accelerations and angular velocities. When looking downwards from a high altitude, we are not able to determine the depth of the scene because of our narrow stereo baseline. Only when we start to move sufficiently we provoke sufficient visual stimulus for depth perception. This phenomenon is also known as Structure from Motion (SfM) in Computer Vision. Without motion, we have vestibular sense (gravity), but no visual stimulus to actually extract gravity from other

accelerations. This leads to an inconsistency between the two sensors and causes alarm signals in our body also known as vertigo (Corke et al. (2007)). Fig. 1.2 depicts such situations.

Conversely, as long as vestibular and visual cues are consistent with each other, we are able to maintain a good sensation of our attitude. For example, as children we spin around our vertical axis to confuse our vestibular system. Professional dancers also do several spins around this axis, but at the same time fix one specific visual reference point. This visual feedback, in fact, subconsciously prevents our vestibular system to get confused. It is essential that, in this example, the transformation between the visual and vestibular sensors is well known due to our sense of position of our joints. In a sense, we have here the situation of a system using fast dead-reckoning sensors for accelerations and angular velocities together with a slower sensor for visual measurements in order to estimate correctly a gravity-aligned pose.

Again, what happens subconsciously in nature, robots (and thus roboticists) have to pay explicit attention to. Knowing the capabilities of nature, there is no doubt, that an optimal merging of different sensor information can be very powerful for robot navigation. In fact, robots have the advantage of being able to use globally consistent signals like GPS or external beacons. Also, their precise memory of previously visited places suggests that they can be superior navigators to human beings or animals. The main advantage of nature is the fast information extraction from a wealth of sensors and the optimal fusion of this information. With vision and inertial sensors as main cues, nature proves to be capable of excellent navigation even in 3D space where there exists no so-called zero position. Such zero positions exist on a ground robot for example, where stopping all actuators is equal to zero velocity. Of course, stopping all actuators on an airborne robot is not an option. This lack of a zero position is particularly challenging for agile and inherently unstable helicopters, since an accurate state estimate for robot control must be available at all times while airborne. Taking nature as example, it is the main goal in our research to achieve long-term navigation for computationally constrained micro helicopters in natural, large and initially unknown environments. With the focus on vision based methods, we seek to identify an appropriate sensor-fusion approach for both, state estimation and sensor self-calibration. We aim at real-world multi-sensor power-on-and-go system.

(a)

(b)

(c)

Figure 1.2: Two situations where a human's sensor fusion capabilities are challenged. In a) it would be literally impossible to perceive any visual depth cues. Thus vision and vestibular system yield inconsistent information. Using extra sensor feeds like touching the ground with the hands mitigate the inconsistent information. In b) even though we do not perceive visually the depth of the whole scene, we have reliable anchor points of the building facade close to us. These visual cues are sufficient for consistent information and vertigo is much less perceived in such situations. In the sequence c) we see a number of different positions to mitigate vertigo (Bronstein (2000)). Vertigo is less present, the more contact points we have to solid ground. In the top picture of the sequence we have numerous sensors close to our vestibular system to estimate our attitude (i.e. extract gravity from our vestibular readings) whereas in the lowest picture, our sensor calibration between the feet sensors and the vestibular system is too imprecise to reliably estimate our attitude. (pictures a) and b): VertigoPhotography, www.very-bored.com, c): (Bronstein (2000)))

1.1 Motivation and Objectives

In contrast to ground robots, for which mobility is limited by their possible interactions with the terrain's 2.5D manifold, micro helicopters exploit their full 3D maneuverability, such that they are not bound anymore to the ground scene. They easily allow to access environments where no humans or other vehicles can get access to, thus, reducing the risk for both the people and the environment. The capability of fast and unobstructed maneuvers above common ground obstacles makes micro helicopters ideal for reconnaissance, search and rescue, and coverage tasks in large, unknown, and possibly cluttered environments. Furthermore, the inherent possibility of helicopters to hover on the spot and quickly act against external disturbances like wind gusts makes them a superior choice for a variety of tasks with respect to fixed-wing alternatives.

For tasks in the domain of surveillance, reconnaissance or coverage there are a number of contradicting requirements to the system. For these applications, it is necessary that the vehicle is able to navigate without relying on a preexisting map or specific assumptions about the environment. This will allow operation in unstructured, unknown, and GPS denied environments. Surveillance tasks involving human beings, typically require inherently safe platforms in order to minimize potential damage on system failures. This requirement suggests very lightweight platforms to minimize the impact energy on crashes. Conversely, the surveillance task requires high quality estimates, and thus often heavy sensor gear for adequate data capture.

In essence, we have contradictory requirements for a variety of different tasks favoring a small and light-weight platform, while at the same time being able to carry large sensor suites over long flight periods. Without doubt, with the advance of energy storage devices, we will be able to mitigate these contradictions. As for now, we focus on platforms small enough to be inherently safe but large enough to carry at least some additional payload (about 200 grams). Such platforms are the agile, inherently unstable multi-copter systems. In contrast to coaxial helicopter configurations, multi-copter systems can counteract well against external disturbances often occurring in outdoor scenarios. Fig. 1.3(a) and Fig. 1.3(b) depict possible surveillance and rescue scenarios for autonomous MAVs.

Despite the fact that micro helicopters have additional Degrees of Freedom (DoF) with respect to ground robots, nature proves the feasibility of navigation in 3D for a variety of different "platforms" even with a minimal sensor suite consisting of visual and inertial sensors only. Developing an approach that lets a micro helicopter navigate in an equally agile manner

(a) Surveillance environment (b) Search & rescue in a desaster area

Figure 1.3: Two possible surveillance a) and rescue b) scenaria for autonomos micro helicopter. Both scenaria require the vehicle to be lightweight and thus inherently safe. Furthermore, globally consistent signals might not be available in urban corridors or in ruins after an earthquake. A viable solution for autonomous micro helicopter navigation is to use lightweight cameras and to apply computationally efficient algorithms to consistently fuse different sensor inputs in order to achieve long-term navigation in unknown environments. (artistic impression, sFly project: www.sfly.org)

is without doubt a highly desired goal. Naturally, the choice of sensors to be used to achieve this goal is not evident and there may not be a unique, best choice. On micro helicopters – and in general on Micro Aerial Vehicles (MAVs) – there are three dominant constraints. (a) Probably most important is that the limited payload budget asks for lightweight sensors which, at the same time, provide rich information. Every 10 grams require roughly 1 Watt lifting power in hover mode of a commercial multi-copter system. (b) The less calculation power it is need, the more lightweight and power efficient the computation unit is. Conversely, the more power the sensor and computation unit requires, the less power available for the system's propulsion. These first two dominant constraints indirectly and directly act on the system's autonomy time. (c) The third dominant constraint is the availability of the sensor signal. While ground robots by definition maintain close proximity to the scene, airborne vehicles may navigate far away from it such that distance measuring sensors or small baseline stereo vision setups fail [1]. Moreover, global information sensors (e.g. GPS) may not receive any signal or may only receive a highly noisy signal.

The inspiration by nature and the bound to these three dominant constraints motivate to focus on vision based navigation approaches including inertial measurements as propiosceptive sensors. Recent advances in both cameras and Inertial Measurement Units (IMU) made it possible to easily fulfill the first dominant constraint of a limited payload budget. It is the second dominant constraint of low power consumption that motivates to find efficient algorithms in order to process the vast data of the camera without the need of a heavy duty computing unit – even when navigating for a long time in large environments. Even though the camera is a passive sensor and hence always yields some information, its limited capabilities in underexposed or visually homogeneous environments are evident. Thus, the third constraint, that a sensor may not always provide good information, motivates to develop an approach where multiple types of sensors can collaborate by merging and/or inter-changing their feeds while flying for increased accuracy and robustness of state estimation.

1.2 Current State of Research

The research in autonomous MAVs is relatively young but advancing very fast. Even though a lot of progress has been achieved in this topic during the past years, we still strive to find small autonomous systems in unknown,

[1] in the case of a small baseline stereo setup, the sensor is reduced to the capabilities of a single camera if the scene is too far away.

cluttered, and GPS denied environments. Only after solving the issues of rudimentary navigation, high level tasks such as autonomous exploration, swarm navigation, and large-scale trajectory planning can be tackled for such systems.

A key problem on an airborne vehicle is the continuous stabilization and control in six degrees of freedom (DOF), that is, attitude and position control. Already Forlanini noted in 1878 the difficulty of this problem when he launched the probably first unmanned helicopter. His version was neither actively stabilized nor steerable. In the following years, it was in particular the military who boosted the advancements of unmanned rotor-crafts. The first multi-copter systems emerged in the beginning of the 1900s and 1907 the first (manned) multi-copter took off (Gyroplane No. I). An extensive overview on the history of airborne vehicles can be found in (Bouabdallah (2007)). Presently, research focuses on autonomous multi-copters where issues of control, navigation, flight time (i.e. energy) and miniaturization are analyzed.

Because MAVs are in general highly unstable and nonlinear systems, a clever combination of sensor equipment and controller must be designed. Most of the approaches model the MAV as two connected ideal subsystems and use a cascaded control structure: one controller for the attitude of the MAV (3D orientation of the helicopter) and one superimposed controller for its 3D position. In general, the attitude controller has to be faster than the position controller because of the vehicle's high dynamics. Often, a simple PD-controller is enough for that purpose, however, also other techniques can be applied for its design (Cai et al. (2008); Castillo et al. (2004); Peng et al. (2007)). Bouabdallah et al. (Bouabdallah and Siegwart (2005)) analyzed the application of two different control techniques, "Sliding-Mode" and "Backstepping", and showed that the latter has particularly good stabilizing qualities. Throughout this dissertation we consider the issue of multi-copter attitude control as sufficiently elaborate. Thus, we use the on-board attitude controller provided by Ascending Technologies (Ascending Technologies GmbH) which is basically a fast paced PD-controller.

Although, with attitude control, it is possible to hold the flying platform in a hovering state, there is no possibility to perceive and correct for any drift caused by accumulated errors. To tackle this issue, exteroceptive sensors (e.g. laser scanners, cameras, GPS, pressure sensors, or ultrasonic sensors) able to measure the vehicle's position are unavoidable. While the (short term) attitude control is probably best tackled using inertial measurements, the position control does not have obvious sensor choices.

The most common approach is to mount a DGPS receiver on the MAV.

By using a so-called inertial/GPS approach, where data from an IMU and GPS data are fused together, the position can be estimated up to a precision of some decimeters. Thus, the MAV can be fully stabilized and controlled (Abdelkrim et al. (2008); Yun et al. (2007)). The two drawbacks of this approach are the necessity to receive a clean GPS signal and the lack of precision of the position estimate. Furthermore, GPS is not a reliable service; its availability can be limited by urban canyons and is completely unavailable in indoor environments. Nevertheless, most approaches for autonomous MAVs have been developed using GPS, sometimes in combination with on-board cameras and IMU. For instance, Ludington et al. (Ludington et al. (2006)) used a Kalman Filter to fuse GPS, vision, and IMU data. Templeton et al. (Templeton et al. (2007)) used a GPS-based flight control system for navigation and vision to estimate the 3D structure of the terrain in order to find adequate landing sites.

Alternatively to GPS, successful results have recently been achieved using laser range finders (Achtelik et al. (2009); Bachrach et al. (2009a,b)). In (Bachrach et al. (2009a,b)), Bachrach et al. used a Hokuyo laser scanner and 2D SLAM for autonomous navigation in a maze. With their platform they won the international aerial robotics competition (IARC), which consisted in entering and exiting from a maze. In contrast to cameras, laser range finders have the advantage to work in textureless environments. Although very appealing, the use of range finders is not optimal since these sensors have a restricted perception distance and limited field of view (typically only in a plane) and are still heavy for MAVs.

The only viable solution for GPS denied environments may be to use vision. The simplest way is to install a number of external cameras with known location and to have them track the vehicle (Altug et al. (2002); Klose et al. (2010); Park et al. (2005)). In the last years, excellent results in this endeavor have been achieved using the motion capture system from Vicon[2]: a system of high-resolution, external cameras that can track the 6DOF pose of one or more helicopters with submillimeter accuracy (How et al. (2008); Michael et al. (2010b); Valenti et al. (2006)). This is achieved by the use of retroreflective markers that are rigidly attached to the vehicle's body and that can easily be tracked by the Vicon's cameras even when all but one camera are occluded. Using the software from Vicon, the tracking of quadrotors is rarely lost, even during extreme situations such as fast maneuvers (velocity of 3.5 m/s, acceleration of 15 m/s^2) (Michael et al. (2010b)). Aggressive maneuvers like multiple flips of the MAV (Lupashin et al. (2010)), docking to an oblique wall (Mellinger et al. (2010a)), or cooperative grasping of objects (Mellinger

[2]http://www.vicon.com

et al. (2010b); Michael et al. (2010a)) have been achieved during 2010 using the Vicon system. Typically, in these installations the number of Vicon cameras for MAV test beds ranges between one and sixteen and the resolution can be up to 16 Megapixels, while the frame rate can reach 375 Hz. Motion capture systems are very efficient and robust for testing purposes and can provide ground truth reference to evaluate other approaches. However, relying on such a system is obviously not suitable for large environments and for missions where the installation of an appropriate infrastructure for external cameras is not feasible. Additionally, it requires an accurate calibration of the camera system.

We would also like to point out the difference between on-board cameras and external (i.e., off-board) cameras: when using external cameras (such as the Vicon), it is not the helicopter that is autonomous but rather the system comprising the external cameras plus the helicopter. It is only in the case of on-board cameras that the helicopter can be considered truly autonomous. Furthermore, a link between the MAV and off-board sensor or/and processor cannot be taken as granted. The benefits of using the camera as an on-board sensor for navigating micro helicopters in large and initially unknown environments are at hand. However, there are still two main issues which we address in this dissertation. Firstly, the amount of information the camera yields is so vast that we need appropriate algorithms to process it on-board a computationally restricted MAV. That is, we are not only aiming at using the camera as a bearing sensor, but as a *real-time* [3] and *on-borad motion* sensor. Secondly, the drift reported in (Ahrens et al. (2009)) impedes long-term navigation in large environments. Moreover, during long missions, sensor calibrations might change favoring further drifts. Thus, we aim at understanding in detail the different drift issues and continuous sensor-calibration possibilities. Since some drift and calibration issues are inherent to a vision-inertial only system, we investigate in a multi-sensor system and its possibilities of continuous sensor self-calibration.

Camera as a Motion Sensor

In general, we consider two approaches using on-board cameras: the first one is tracking a known, fixed object (like artificial markers, or user-specified points), which implicitly solves the matching and relocalization problem because the markers are already known as well as their relative 3D position;

[3] The real-time capability of the camera as a *bearing* sensor is mainly given by its framerate. For *motion estimation*, the images undergo computational complex tasks and state-of-the-art still lacks of providing real-time general camera motion estimates on computationally constrained platforms.

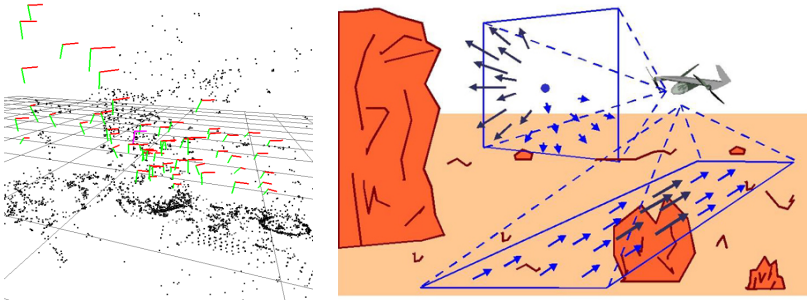
the second approach is extracting distinctive natural features – whose 3D position is not known a priori – and use them for both motion and structure estimation, and relocalization (i.e. visual SLAM, visual odometry). Many different approaches have been developed within the first category (Hamel et al. (2002); Proctor and Johnson (2004)). Hamel et al. (Hamel et al. (2002)) implemented a visual trajectory tracking method to control a MAV with an on-board camera observing n fixed points. A similar approach was developed by Cheviron et al. (Cheviron et al. (2007)), who additionally fused the vision with IMU data. Another possibility is to have the MAV tracking a leading MAV and maintain fixed relative position and orientation. Chen and Dawson (Chen and Dawson (2006)) implemented it by tracking some coplanar points on the leading vehicle and using homography to estimate the relative pose. In contrast, Wenzel et al. (Wenzel et al. (2010)), used four light sources on a ground robot and used homographies to perform autonomous take-off, tracking, and landing on the moving ground robot.

To the best of our knowledge, very little work has been done within the second category, that is, extracting natural and initially unknown features for controlling the helicopter. These works are described in and (Artieda et al. (2008)) and (Ahrens et al. (2009)). The work of Artieda et al. (Artieda et al. (2008)) describes the implementation of a visual SLAM system based on an Extended Kalman Filter (EKF) for UAVs, but their visual SLAM is used only for mapping and not for controlling the helicopter. Therefor, the work that is probably closer to ours is that from Ahrens et al. (Ahrens et al. (2009)). Based on the monocular EKF-SLAM algorithm of Davison et al. (Davison (2003)), they built a localization and mapping framework that provides an almost drift-free pose estimation. With that, they implemented a position controller and obstacle avoidance. However, due to the simplification they used to speed up their feature-tracking algorithm, a non-negligible drift persists. In addition, they used the Vicon motion capture system to control the aerial vehicle. Therefor, the output of the SLAM-based localization system was not used for the vehicle stabilization. Another problem is represented by the choice of an EKF-based visual SLAM algorithm that is particularly sensitive to outliers and requires high computation power for large environments. In fact, long-term navigation in large environment is computationally infeasible with such an approach since with N features, the complexity is $O(N^2)$ at its theoretical best. Consequently, none of these approaches are on-board implementations. It is one of our goals to render the camera as an on-board and real-time 6DoF pose sensor for full MAV control. As one of the most modern, high-performing systems, we choose to tailor PTAM (Klein and Murray (2007)) to the general needs of a computationally limited MAV

platform.

Most vision-based works developed so far concentrated on specific, individual tasks only such as take-off, landing, hovering, or obstacle avoidance. For instance, Saripalli et al. (Saripalli et al. (2002)) and Mejias et al. (Mejias et al. (2006)) focused on autonomous landing using only vision or vision and IMU, respectively. In particular, Mejias et al. (Mejias et al. (2006)) used vision to identify a good landing spot and reach it as fast and safely as possible. Most vision-based controllers for hovering-purposes only have been implemented by means of optical flow (Herisse et al. (2008); Zufferey and Floreano (2006)). Herisse et al. (Herisse et al. (2008)) used an optical flow based PI-controller to stabilize a hovering MAV. They also implemented an automatic landing routine by analyzing the divergent optical flow. However, optical-flow-based hovering is affected by drift since it computes the relative displacement of the features between the last two frames (i.e. the body-velocity) and not their absolute motion, for example, with respect to the initial frame. This also holds for visual-odometry-based implementations, where the pose is estimated by considering the feature displacements between two consecutive images (Fowers et al. (2007)). Optical flow, however, turns out to be extremely useful for obstacle avoidance and wall or terrain following because of the reactive nature of these tasks. Once a vehicle is stabilized in its pose, reactive controllers can be applied to steer specific behaviors of the system. Note that, on ground vehicles, the pose stabilization is straight forward since the system is stable with zero actuator input. Thus, reactive commands can directly be applied. Conversely, airborne vehicles lack of this "zero-position" and first need to actively be stabilized in the 3D space in position and attitude. Thus, reactive controllers without having a proper and continuous state estimate is insufficient for navigation of such vehicles.

Based on optical flow, some biologically inspired control algorithms have been developed in this endeavor, as reported in (Garratt and Chahl (2008); Hrabar et al. (2005); Ruffier and Franceschini (2004); Zufferey and Floreano (2006)). For example, a wall-collision avoidance was developed by Hrabar et al. (Hrabar et al. (2005)), who implemented a strategy to navigate in urban canyons. They used optical flow from two cameras placed on both sides of the vehicle. In addition, they used a front-looking stereo camera to avoid oncoming obstacles. In our previous work (Zingg et al. (2010)), we also used optical flow to control a quadro-copter to remain in the center of a corridor. In (Garratt and Chahl (2008)), Garrat and Chahl used optical flow to perform terrain following with a co-axial helicopter. Fig. 1.4(a) and Fig. 1.4(b) depict the two different approaches of map-based (using structure from motion) and map-less (using optical flow) approaches.



(a) (b)

Figure 1.4: In a) a map-based approach for camera pose estimation is shown. Computationally inexpensive algorithms use only distinct camera frames with their associated features to build a map and localize the current camera frame within. In b) we show a map-less approach. Optical flow might be well used for reactive maneuvers. However, for airborne vehicles a reactive control is not sufficient, as they need continuous pose control in order to navigate stable in 3D space. (pictures: a) PTAM, b) Centeye, Inc)

Even though some of the approaches mentioned above work on-board, none have been used to control the MAV in all 6DoF. This suggests that optical flow might be used for reactive maneuvers, but not for full MAV control. To (velocity) control such agile systems as micro helicopters, a reactive control is, indeed, not sufficient. In fact, since a flying micro helicopter has no “rest” or “zero position“ in the state space, it is crucial to continuously have knowledge of the full state of the MAV as accurately estimated as possible. Reactive commands as ”too close“, “too far away“ tend to oscillate and eventually become unstable. Despite these issues, this dissertation shows a method to velocity-control an MAV based on an optical flow approach using inertial constraints.

Drifts and Sensor Self-Calibration

For long-term MAV navigation in large environments, drifts in the MAV pose estimate play an important role for the proper control of the vehicle. Inertial measurement alone are known to yield rapidly drifting velocity and position estimates. It is common to mitigate this drift adding extra sensors such as cameras.

Fusing IMU data with a vision sensor is a well studied topic in the literature. The metric scale is by nature not recoverable in single-camera approaches and a failure results in the need of re-initialization. With known extrinsic parameters, an IMU can both recover the metric scale and bridge short failure periods of the vision part. However, most works in the literature focus either on the calibration of these two sensors or on the pose estimate of the robot assuming known calibration parameters. As analyzed in (Corke et al. (2007)), there are two different ways to combine inertial and visual measurements for absolute scale structure and motion estimation, which are called *loosely-coupled* and *tightly-coupled*. The loosely-coupled approach treats the inertial and vision units as two separate modules running at different rates and exchanging information, while the tightly-coupled approach combines them into a single, optimal filter. Among the loosely-coupled approaches are the works of (Armesto et al. (2004, 2007); Corke (2004); Gemeiner et al. (2007); Mourikis et al. (2009); Roumeliotis et al. (2002); Weiss and Siegwart (2011)), while among the tightly-coupled ones are those of (Baldwin et al. (2009); Chroust and Vincze (2004); Huster et al. (2002); Jones (2009); Jones and Soatto (2010); Kelly and Sukhatme (2011); Lupton and Sukkarieh (2008, 2009); Qian et al. (2002); Strelow and Singh (2003)).

Already early work (Strelow and Singh (2002)) shows the improvements on pose estimation combining vision and inertial sensors. In (Corke (2004)), Corke uses a stereo camera and an IMU to improve feature matching. The velocity of the system is then computed by fusing the two measurements in a complementary filter. In (Armesto et al. (2004)), Armesto et al. also use a stereo camera and multi-rate data fusion with an EKF or an Unscented Kalman Filter (UKF). In their following work (Armesto et al. (2007)), they replace the stereo camera with a single camera and show 6DoF fusion of IMU and vision on a robot arm. They also compare the performance between EKF and UKF. The scale is recovered from the known dimensions of the observed patterns. Thus the working area is limited to the area where the pattern can be observed. Moreover, the authors focus on the pose estimate and the inter-sensor calibration is assumed to be known. The authors state that the UKF approach yields better results at the cost of higher calculation power. We set a high priority on speed of the algorithm as we aim at real time full 6DoF state estimation. Thus we implemented our decoupled solution in an EKF framework only.

Real-time vision-inertial navigation using an iterated EKF has been shown by Strelow in (Strelow (2004)). The complexity grows at least quadratically with the number of features and is not suitable for large scale robot navigation. A more efficient approach is presented in (Roumeliotis et al. (2002))

and (Mourikis and Roumeliotis (2007); Mourikis et al. (2009)), which considers pairwise images for visual odometry and fuses the output afterwards with inertial measurements in an EKF. In this work, the authors also use an additional altimeter for solving the unknown scale of the vision algorithm, presenting an EKF approach which is linear in the amount of features and quadratic in the number of included camera poses. All these approaches, however, do not include the inter-sensor calibration nor the possibility of augmenting the system with additional sensors or exchanging the vision part with a readily available solution.

Most closely related to the present dissertation is (Mirzaei and Roumeliotis (2008)) and (Kelly and Sukhatme (2011)). In (Mirzaei and Roumeliotis (2008)), Mirzaei and Roumeliotis present a Kalman Filter (KF) based calibration solution to estimate rotation and translation between a camera and an inertial sensor mounted on a rigid body. They retrieve the scale from a feature pattern with known dimensions. We consider this work as solid foundation for (Kelly and Sukhatme (2011)) and work in this dissertation. In (Kelly and Sukhatme (2011)), the authors extended the calibration method to the observation of an a priori unknown static structure. This allows to estimate the calibration between IMU and camera without any additional equipment. In fact, since their work estimates the gravity vector in the vision frame [4], it is a complete vision-inertial, self-calibrating EKF based navigation system. Their approach allows to continuously recover the roll and pitch angular offsets of the vision-inertial system with respect to an earth-fixed reference frame even if the initial anchor landmarks are not visible anymore. The underlying visual EKF SLAM approach, however, impedes the use on MAVs in large scale environments because of the rapidly increasing computational complexity.

The work of (Jones (2009)) and (Pinies et al. (2007)) have the same spirit as the work of (Kelly and Sukhatme (2011)). The authors couple the IMU and camera information tightly in a single EKF SLAM framework. This is in essence similar to the approach followed by (Kelly and Sukhatme (2011)), however with the obvious extension to new features. From the filter perspective, these approaches are robust as they take all cross-correlations between the observed feature positions and motion states into account. However, as mentioned beforehand, the computational cost of EKF SLAM is $O(N^2)$, with N being the amount of features. This issue prevents long term runs or requires sub-mapping at the cost of cross-correlations. In (Bryson et al.

[4] We refer to the *vision frame* as the frame with respect to which the camera pose is estimated in the black-box vision framework. The authors in (Kelly and Sukhatme (2011)) refer to this frame as *world frame* which they clearly state is *not* a gravity-aligned frame. In this dissertation, when we refer to a *world frame* we talk about a gravity-aligned frame unless otherwise noted.

(2009)), the authors fuse a camera, IMU and GPS in an EKF SLAM framework. They mention a state dimension of 10^4. This is clearly not computable on a robot with reduced calculation power. Also, the inter-sensor calibration is again assumed to be known.

In (Strasdat et al. (2010)) the authors illustrate the requirement of many features in order to do robust pose estimation and thus favor a key-frame based SLAM approach in most scenarios. The latter approach runs at linear time in the number of key-frames (navigating in a previously visited area). This dissertation presents the possibility of decoupling the self-calibrating navigation system from the computationally expensive EKF-based vision part, and notably replace the vision part by a (possibly closed-source) key-frame based approach. Maintaining constant computational complexity, the loosely-coupled system presented here becomes scalable to large environments. Despite the sacrifice of cross-correlations as aforementioned (we only consider the final calculated camera pose) we show that the system still works reliably both, in theory as well as in practice.

1.3 Contributions

This dissertation investigates in methods for vision based navigation for micro helicopters. The main goal is an approach which is not only able to provide long-term navigation of the vehicle even in areas without global positioning signals, but is also able to cope with the in-flight calibration of a variety of different sensors and with their different availability during flight.

This long-term goal requires intensive research in a multitude of areas. This dissertation presents the following core contributions:

1. *Development of vision algorithms that render the camera a real-time on-board sensor for pose estimates on computationally constrained airborne vehicles.*

2. *Development of a multi-sensor fusion framework capable to estimate at the same time the vehicle's pose for control and the (inter-)sensor calibration for continuous operation.*

3. *Theoretical observability analysis of this framework to identify unobservable modes and their connections in the state space.*

4. *Implementation and testing of the (linearized and discretized) framework in simulation, on real platforms and in real-world scenarios*

With these four key-points in focus, several side-paths are explored and discussed.

While the hardware design and concept of micro helicopters evidently plays a major role to achieve our goals, in this dissertation, we focus on the algorithmic side of the problem statement. The hardware design has been studied for various different implementations such as (Achtelik et al. (2011a); Bouabdallah and Siegwart (2007); Oosedo et al. (2010); Rawashdeh et al. (2009)). Thus, this topic is considered as out of the scope of this work and commercially available platforms from Ascending Technologies, GmbH [5] are used. Together with the system concept and design, developers provide a low level control scheme for attitude stabilization (Achtelik et al. (2011a); Bouabdallah (2007)). A number of different control strategies have been presented in particular in (Bouabdallah (2007)) but also in (Bloesch et al. (2010); Mellinger and Kumar (2011); Schafroth (2010)). We consider the different control strategies as dependent on the platform. Therefor, we focus on the sensor-dependent, but not on the platform-dependent state estimate of the vehicle. This estimate is general and can be used in the different control approaches discussed in literature.

1.3.1 Camera as a Motion Sensor

With respect to sensors, the focus of this dissertation is on using a standard camera as an on-board and real-time motion sensor for full pose control of a micro helicopter. In particular, we aim at obtaining real-time pose or body-velocity measurements from this sensor mounted on-board the platform. While such algorithms are state-of-the-art for off-board platforms (Civera et al. (2010); Davison et al. (2007); Klein and Murray (2009); Newcombe et al. (2011)), it is still a great challenge to retrieve such information on a computationally and payload constrained, flying vehicle. We present a contribution by modifying a state-of-the-art monocular pose estimator such that it meets our requirements for a pose measuring sensor on MAVs. Besides increasing speed and robustness of the algorithm by modifying its concepts, we analyze its new behavior with respect to drift. We show that, despite our modifications to gain constant computational complexity, the monocular pose estimator keeps all drifts very low. The outcome is a modular "sensor" capable of estimating the MAV's 6DoF pose (up to scale) at 20 Hz on a standard 100 gram ATOM 1.6 GHz single core board. This contribution advances the state-of-the-art such that visual 6DoF pose estimation becomes possible on computationally and payload constrained platforms.

[5] www.asctec.de

Monocular pose estimation is based on a map (usually a 3D point cloud) from which the camera pose is retrieved. A core issue is the possibility of "losing" this map by either sudden light changes, low-contrast regions, occlusions or mismatches. In these cases, position drifting, short term velocity control can bridge the time needed for re-initializing a new reference map. Body-velocity of a camera is generally estimated by Optical Flow (OF) approaches. In contrast to the monocular pose estimation algorithms, OF methods already found application on micro flying vehicles with on-board computation (Beyeler (2009); Zufferey et al. (2007, 2010)). State-of-the-art optical flow approaches have the drawback that special cameras such as 1D black-and-white CMOS chips are used in order to reduce computational complexity. These sensors require highly textured environments and are specialized for OF calculation. Furthermore, the same camera cannot be used to retrieve additional information needed for visual pose estimation, human remote operation, mapping, coverage, or reconnaissance tasks.

In this dissertation, we present a contribution by developing an OF approach capable of velocity control the MAV and self-calibrating the sensor suite using a standard global shutter WVGA camera and an IMU. The same camera has been used for the monocular pose estimation mentioned above and can be further applied to provide visual feedback to operators and in general to higher level tasks. This saves cost, power and weight. Our OF approach uses inertial constrains to estimate the vehicles body-velocity at 40 Hz on a standard 100 gram ATOM 1.6 GHz single core board.

1.3.2 State Estimation and Sensor Self-Calibration

The main contribution of this dissertation is the theoretical development and practical implementation of a multi-sensor self-calibrating state estimation for micro helicopters with limited payload and computation budget. We focus on vision and IMU as main sensors for locally stable maneuvers since these sensors are both, lightweight and low on power consumption. Yet, they yield sufficient information to achieve the task. We distinguish our work from the state-of-the-art by aiming at a *self-calibrating power-on-and-go* system for long-term navigation in large environments. Current research provides limited scalability for visual-inertial systems (Kelly and Sukhatme (2011)).

We advocate that all processes should have constant computational complexity. Only this ensures scalable operation in large environments. Hence the approach in (Kelly and Sukhatme (2011)) is not an option. We provide an approach which decouples all sensors and refers them to one common reference frame. This makes our method not only constant in computational

complexity but also makes it inherently versatile and modular for different sensor setups. As a second requirement, drifts in the whole system with respect to an (Earth) fixed frame have to be identified and compensated for in order to ensure long-term operation. Drift in either the roll or pitch angle of the vehicle's state estimate for example, would lead to an estimate which is not aligned to gravity anymore and would cause the aerial vehicle to crash. Hence, it is very important to identify and compensate for such drifts. Here, we analyze the addition of different sensor types to our modular approach in order to eliminate different drift sources affecting visual position, attitude, and scale drift.

Each sensor addition, including the camera, introduces additional inter-sensor calibration states to the system. Also, the IMU itself has already intrinsic calibration states such as the acceleration and gyroscope biases. We investigate in detail which of the introduced (inter-)sensor calibration states are observable. We introduce these states to our system and show that we can generally avoid any calibration procedure. This makes our approach a possible choice for power-on-and-go systems. For example, we cannot only show the obvious result that adding a magnetometer renders the absolute yaw of the system observable, but we can also show that it is not necessary to apply any model of the earth's magnetic field. In fact, as long as we consider an arbitrary but constant azimuth as north, we show that one can estimate the elevation of the magnetic field vector rendering a magnetic field model superfluous. Furthermore, we investigate on unobservable modes in a given sensor setup and draw conclusions on how to handle new measurements from additional sensors during flight. This is particularly interesting for questions regarding which states will be affected with a sudden sensor failure or additional incoming information. Such scenaria occur for example on transitions from outdoor to indoor where GPS signal availability changes.

Besides this theoretical analysis, we implement our approach for simulations and for testing on a real platform. This step is essential to complete our main contribution since the implemented system represents a linearized and discretized version of the theoretically analyzed one. We present extensive simulations on convergence behavior and on the behavior when different states are initialized off their true value. The final system is then implemented on a real platform and both tested quantitatively on ground truth data and qualitatively in long-term operation in a large-scale environment. Altogether, we bring the theoretical system developed on a solid theoretical basis to a real system, tested and evaluated out of the lab in real-world scenaria.

1.4 Structure of this Dissertation

This introduction is followed in Chapter 2 by a presentation of our methods to render the camera a real-time on-board pose or body-velocity sensor. In the first method – a map based method – we highlight our modifications in a state-of-the-art monocular pose estimation algorithm, introduce the notions of drift in such a monocular system and analyze our modified method with respect to position, attitude and visual scale drift. The second method – a map less method – renders the camera a body velocity sensor based on optical flow and inertial constraints. This will allow us later to velocity control the helicopter in order to mitigate map-related issues of the first method. We do not apply any sensor fusion at this stage yet.

Once we decoupled the camera as a general motion sensor, we concentrate on the theoretical analysis of our filter based sensor fusion framework in Chapter 3. We start the analysis with a core sensor-suite consisting of an IMU and a camera only. After, we gradually introduce more (drift compensating) sensors and will end the chapter on a higher level of abstraction where we consider the concatenation of general sensor *types*. The essence of this chapter provides a deep understanding of the (un-)observable modes of a system with a given sensor setup. Furthermore, it shows that we can generally omit prior (inter-)sensor calibration procedures.

In Chapter 4 we test our previously theoretically analyzed system in both simulations and on the real hardware in real-world scenaria. These tests reveal different sensitivities of the system and show a functioning, self-calibrating real-world power-on-and-go airborne platform.

The dissertation itself is concluded in Chapter 5, where our results are put into perspective relative to high level applications such as environmental mapping and coverage as well as to future work.

Chapter 2

Camera as a Motion Sensor

This chapter introduces the camera as a valid on-board and real-time *motion* sensor on computationally and payload constrained micro helicopters for long-term navigation in large, initially unknown environments. In contrast to other (visual) motion estimation approaches, we do not require any a priori information on the environment, nor do we need external aids such as an external tracking system in order to obtain reliable information for MAV control. In particular, we discuss two approaches: a map based approach where we use the camera as a real-time pose sensor and a map-less approach, where we use the camera as a real-time body-velocity sensor. Throughout this dissertation, we refer to a map as an entity with information about the environment which is automatically built by the autonomous vehicle and at the same time used by the same for localization in its 6DoF pose (position and attitude). In our case, if not otherwise mentioned, this map is a 3D point-cloud built by triangulation of distinctive image points detected in different camera frames.

Section 2.1 as the first part of this chapter discusses the map based approach using the camera as a 6DoF pose sensor and is organized as follows. In Section 2.1.1, we summarize the original SLAM algorithm that has been used and explain the reason of this choice for our approach. In Section 2.1.2, we discuss our modifications and analyze their influences with respect to the original code [1]. In Section 2.1.3 we summarize briefly our results of a feasibility study published in (Bloesch et al. (2010)). This study shows all

[1] Section 2.1.1 and Section 2.1.2 are part of the work in (Weiss et al. (2011b))

basic maneuvers such as hovering, set point and trajectory following, vertical take-off, and landing with only one single camera as exteroceptive sensor. The study shows real experiments demonstrating that the camera poses are computed sufficiently fast for stable MAV maneuvers and that the vehicle can fly autonomously in an unknown and unstructured environment. In this chapter, we are interested in the validity of the visual signal for MAV control as such and we are not yet applying sophisticated sensor fusion strategies.

Section 2.2 as the second part of this chapter discusses the map-less approach (presented in (Weiss et al. (2012b))) using the camera as a 3DoF body-velocity sensor and is organized as follows. In Section 2.2.1, we describe our map-less inertial-optical flow based approach. It recovers the arbitrarily scaled camera velocity which can then be used for MAV velocity control. We show simulation results and tests on the real hardware in Section 2.2.2 Since this approach needs careful sensor fusion strategies in order to be used for MAV control, we provide results on the real platform in 4.2.7 after discussing the underlying fusion theory in 3.4. The whole chapter is concluded in Section 2.3.

2.1 Map Based Approach: The Camera as a Pose Sensor

In this first part of this chapter, we present the map based approach where we use the camera as a pose estimator. This approach is base on the visual SLAM algorithm PTAM (Klein and Murray (2007)), which – as recently shown in (Strasdat et al. (2010)) – is both, computationally less expensive and more robust than filter-based visual SLAM implementations. The computational advantages emerge from the fact that the approach in (Klein and Murray (2007)) is a *key-frame*-based algorithm similar to structure-from-motion methods in computer vision. A *key-frame* represents a distinct camera frame associated with a 6DoF pose and corresponding 3D features. A bundle of sparsely placed key-frames with their associated features represent the map in which the camera is localized at every camera frame. A batch optimization ensures the consistency of all feature positions and key-frame poses. In a previously mapped area, only the tracking task for the current camera frame within the current map has to be performed, thus, only the addition of new key-frames to the map (i.e. exploring new environment) is computationally critical.

We modify the original visual SLAM algorithm to a visual odomerty framework and examine its drift behaviors with respect to the original code.

We also provide computation timings regarding the tracking and in particular the key-frame addition.

2.1.1 Description of the Visual SLAM Algorithm

The approach presented in this section is based on the visual SLAM algorithm PTAM of Klein and Murray (Klein and Murray (2007)) in order to localize the MAV with a single camera in 6DoF (see Fig. 2.1). In summary, Klein and Murray split the simultaneous localization and mapping task into two separate threads: the tracking thread and the mapping thread. The tracking thread is responsible for the tracking of salient features in the camera image, i.e., it compares the extracted point features with the stored map and thereby attempts to determine the pose of the camera. This is done with the following steps: first, a simple motion model is applied to predict the new pose of the camera. Then the stored map points are projected into the camera frame and corresponding features are searched. This step is often referred to as data association. Next, the algorithm refines the orientation and position of the camera such that the total error between the observed point features and the projection of the map points into the current frame is minimized. Thereby the mapping thread uses a subset of all camera frames – also called *key-frames* [2] – to build a 3D-point map of the surroundings. The key-frames are selected using some heuristic criteria which are based on a distance measure and on the visibility (i.e. overlap) of the last key-frame. After adding a new key-frame, a batch optimization is applied to refine both, the map points and the key-frame poses. This attempts to minimize the total error between the reprojected map points and the corresponding observations in the key-frames. In the computer vision community, this procedure is commonly referred to as bundle adjustment. The tracking and the mapping threads operate on the original image as well as on down-sampled versions of it – so called pyramidal levels. On each key-frame creation, features extracted on different such levels are introduced to the map. A pyramidal level prediction procedure in the tracking thread searches the features stored in the map in the predicted pyramidal level of the current image. If a feature extracted at level 1 (first down-sampled image) and added to the map, it will be searched for in level 0 (original image) if the helicopter moves away from it.

There are several important differences between the key-frame-SLAM algorithm considered here and the standard filter-based approach (e.g. (Davison (2003))). Klein and Murray's algorithm does not use an EKF-based state

[2] Here, a key-frame consists of the camera image including sub-sampled pyramidal versions of it, a 6DoF pose and associated 3D map points.

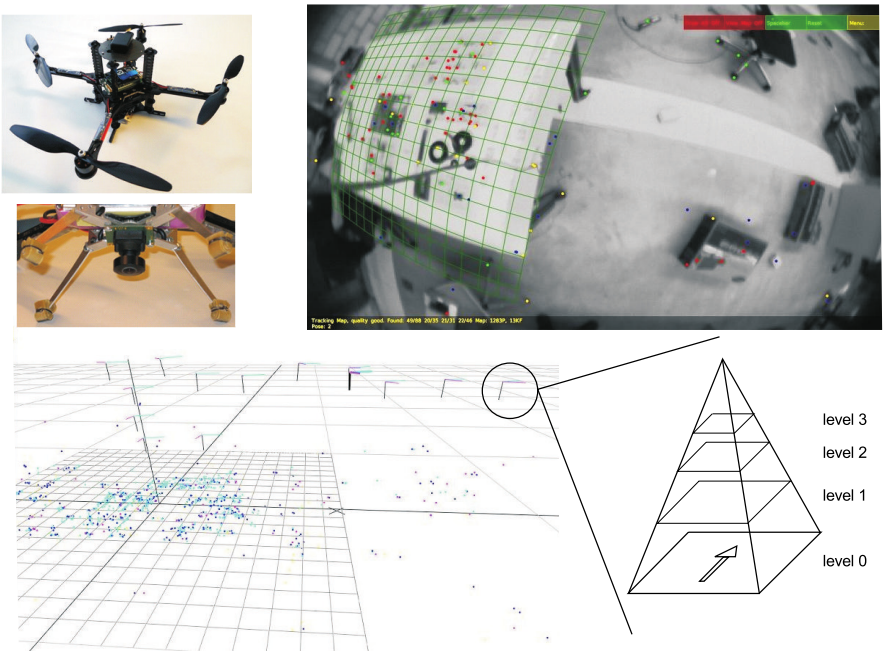


Figure 2.1: The top-left picture depicts our vehicle (the *Pelican*) from Ascending Technologies. Beneath it, the on-board-mounted camera. The top-right picture is a screen-shot of Klein and Murray's visual SLAM algorithm. The tracking of the FAST corners can be observed. This is used for the localization of the camera. In the bottom picture, the 3D map built by the mapping thread is shown. The 3-axis coordinate frames represent the location where new key-frames were added. Each key-frame consists of a 6DoF camera pose, the camera image and its down-sampled pyramidal levels schematically depicted on the bottom right. (Weiss et al. (2011b))

estimation and does not consider any uncertainties, neither for the pose of the camera nor for the location of the features. This considerably reduces computational complexity. Considering the uncertainty of the state could ease the data association process. The lack of modeling uncertainties, however, is compensated by the use of a large amount of features and the global and local batch optimization. Therefor, despite using a fixed area for feature matches, the algorithm is still able to track efficiently the point features and to close loops up to a certain extent. This makes the algorithm extremely fast and reliable, and the map very accurate. As demonstrated in a recent paper by Strasdat et al. (Strasdat et al. (2010)), key-frame SLAM outperforms

filter-based SLAM.

The main advantage of the thread splitting lies therein that both the mapping and the tracking thread can run at different frequencies. Thus, the mapping thread is able to apply a much more powerful and time-consuming algorithm to build its map. Simultaneously, the tracking thread can estimate the camera pose at a higher frequency. This strongly improves the performance. Compared to frame-by-frame SLAM (such as filter based EKF SLAM), the algorithm of Klein and Murray substantially reduces computational complexity by not processing every single image for the map. Particularly when using a camera with a wide field of view, consecutive images often contain a lot of redundant information. In addition, for example, when the camera is moving very slowly or if it stays at the same position, the mapping thread rarely evaluates the images and, thus, requires only very little computational power.

The fact that we are considering the camera down-looking increases the overlapping image portion of neighboring key-frames, so that we can loosen the heuristics for adding key-frames to the map. In addition, once the MAV has explored a certain region, no more new key-frames will be added within the region boundaries. Conversely, when exploring new areas the global bundle adjustment can become very expensive. The increasing computational complexity limits the number of key-frames to a few hundreds on our platform.

Another strength of this monocular SLAM algorithm is its robustness against partial camera occlusion. The large amount of features considered in each camera frame allows large occlusions while the pose estimate is still accurate enough to sustain stable MAV control.

An intricate hurdle when using a monocular camera is the lack of any metric depth information. Therefor, the algorithm must initialize new points based on the observations from more than one key-frame. This could motivate the use of a stereo camera. However, for a stereo camera to bring any further advantage, the observed scene must be within some range according to the stereo-baseline, otherwise a single camera will yield the same utility. Closely linked to this problem is the arbitrarily scaled map inherent to monocular approaches. We did initial studies on recovering the metric scale in (Nutzi et al. (2010)) and will go into further detail in Chapter 3.

2.1.2 Adaptations: From Visual SLAM to Onboard Visual Odometry

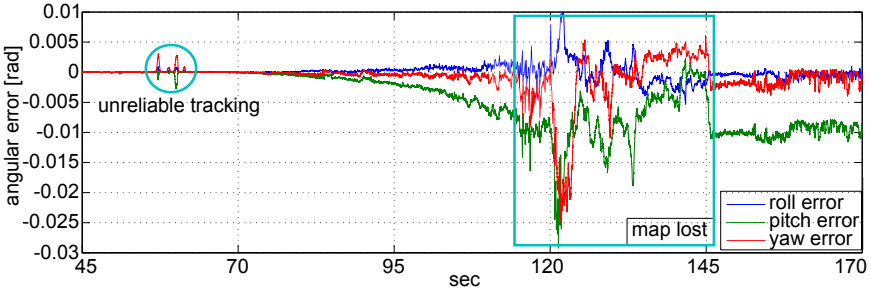
In order to analyze the behavior of the visual algorithm we ran several tests in a Vicon-motion-capture system to compare it to ground truth. Apart from the algorithm speed, there are three particular types of behaviors we are interested in when using the estimated pose as input for a controller: the scale drift, the map or pose drift, and localization failures.

In the tests in the Vicon-motion-capture system, we identified a non-negligible rotational map drift of the original visual SLAM algorithm so that the reference coordinate frame is no longer correctly aligned with gravity. Rotations around the x-axis and y-axis are very problematic because the error between the reference value and the actual position is no longer correctly decomposed into its x, y, and z values. This can lead to instability of the MAV if not considered. In Figure 2.2(a) the rotational drift for a linear trajectory is presented. Since the movement occurs in the x-direction, the pitch-angle estimation is mostly affected by the drift. The same plot also reveals when the tracking thread lost the map and when unreliable tracking occurred. Note that after a loss of the reference map, the algorithm may recover but the visual reference coordinate frame may be shifted in position and attitude. This is shown in figure 2.2(a) at time t=145. These changes are crucial to observe and identify for stable long-term MAV navigation. In Fig. 2.2(b), the resulting deviations in position are depicted. The drift in negative pitch for example leads to a higher estimated z-value than it is in reality. Of course, the more thorough a camera calibration is and the more widely spread the features are, the less are the drifts.

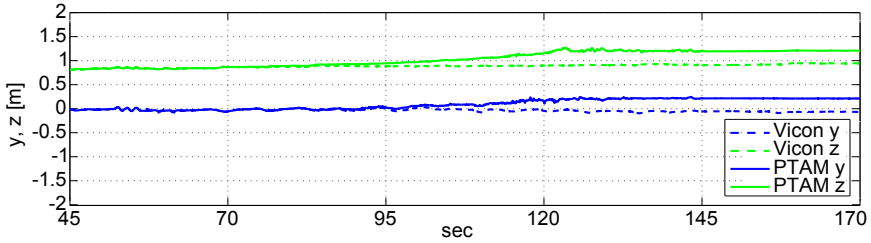
Common to all drifts (scale, position, and attitude drift) is their spatial nature. While observing the same features, the visual pose estimate does not suffer from any drifts. This is the case in hover mode or by only performing movements with a bounded action radius maintaining visibility of some anchor features.

In the original monocular SLAM framework PTAM, drifts can be minimized by applying navigation patterns with recurrent loop closures. A full SLAM system is, however, computationally too complex for large environments. In the following, we detail our modifications and show their influence on performance with respect to drifts, failures, and timings. The changes enable on-board operation at 20 Hz on an on-board Atom 1.6 GHz computer even in self-similar outdoor scenes.

The original code has been ported to be compatible to the Robot Oper-



(a)



(b)

Figure 2.2: a) Map drift represented in angular errors. Note that, as we move in positive x-direction, the angular error in pitch is most affected. From t=115 s until t=145 s the map is lost and the data is corrupted. After, the map is recovered again but at a different attitude. Note that this plot also reveals unreliable tracking parts as shown around t=60 s. b) Due to the angular drift, the position experiences an error as well. For example, the drift towards negative roll leads to a higher z-position compared to reality. The wrong angle estimation eventually leads to instability of the MAV because of misalignment to the gravity vector. (Weiss et al. (2011b))

ating System[3] such that:

- the input image to be processed is taken from a common image node and a processed verification image including the features found, is published. This enables the user to handle PTAM on an embedded system without human-machine interfaces.
- the 6DoF pose is published as a pose with a covariance estimation calculated from PTAM's internal bundle adjustment. This is crucial for later filtering steps discussed in 3.1
- the visualization of camera key-frames, trajectory and features is ported to RVIZ such that visualization and verification can be done on a ground station, if necessary.

Key-frame Handling

In PTAM, the map is defined as a set of key-frames together with their observed features. In order to render the computational complexity constant, we set a maximum number of key-frames retained in the map. If this number is exceeded, the key-frame furthest away (Euclidian distance) from the current MAV pose gets deleted along with the features associated to it. If the maximum number of retained key-frames is infinite, then the algorithm is equivalent to the original PTAM. Conversely, if we set a maximum of 2 key-frames, we obtain a visual odometry framework. Values in between approximate a sliding window approach. Naturally, the larger the number of retained key-frames, the smaller the drifts, but also the larger the computational complexity for the bundle adjustment step. Naive bundle adjustments have a complexity of $O((m+n)^3)$ whereas sparse bundle adjustments have complexity $O(m^3 + mn)$ with n features and m key-frames. This is a large improvement since usually $n \gg m$. Nevertheless, in both cases, the complexity is cubic with the amount of retained key-frames.
There are mainly two different approaches to define the drift in the visual framework. One approach is to consider it as a bias term P_{bias} on the visual pose estimation $\hat{P}_v^c$

$$P_v^c = P_{bias} * \hat{P}_v^c \tag{2.1}$$

where $P_i^j = [R\ T]$ is a transform in rotation R and translation T from frame i to frame j and P_v^c is the true camera pose with respect to the vision frame. This definition keeps the frame, in which the camera pose is estimated

[3] www.ros.org

(i.e. the vision frame) fix and changes the bias terms. A potential misalignment P_v^w between a world frame and the vision is thus constant. The second approach treats the camera pose estimation as unbiased but allows the vision frame to drift with respect to a world frame. An initial misalignment P_v^w between a world frame and the vision is thus not constant. In this dissertation we use the latter definition for easier discussions in the following chapters.

We analyze the drift behavior of P_v^w for different numbers of retained key-frames. Considering (Sibley et al. (2009)) we expect an asymptotically improvement towards the original code the more key-frames we keep. In (Sibley et al. (2009)), the authors only optimize the map in the vicinity of the current position of the robot arguing that changes on more distant map elements become sufficiently small such that they can be neglected. The selection process, which map part is to be optimized is non-constant in computational complexity and depends on the map size. In our approach we gain constant complexity by eliminating distant map elements (i.e. key-frames) avoiding the selection process completely. Furthermore, since distant map elements have a negligible influence in the optimization step (Sibley et al. (2009)) their deletion does barely affect the optimization result.

Fig. 2.3 shows the angular drift between the vision and world frame. Be aware of the different scale in the plots. We note two major disturbances — we call them failure modes — one around t=7 s and one around t=19 s. There, all runs, including the original code, showed pose estimation issues due to dynamic features (i.e. objects providing salient points got moved away by the MAVs turbulences). This renders the data-set challenging but more close to real outdoor environments. Because of these failure modes, the plots are slightly more difficult to read: a large final error does not reflect large drift. Drift behavior is best observed before, between, and after the failure modes. Although all runs show similar attitude drift behavior, the ones retaining fewer key-frames have the tendency to drift slightly faster. The yaw drifts are the slowest and least affected by failure modes. This may be due to the coupling of roll and pitch with the uncertain feature depth estimation and the coupling of yaw with the more certain xy feature position. The fact, that the failure modes are well presented in spikes in these graphs will be used in Section 3.1.2 to determine which visual measurements are sufficiently reliable to fuse with the inertial inputs for MAV pose estimation and sensor-(self-)calibration.

We extract the pure position drift shown in Fig. 2.4 by un-rotating the MAV pose by its attitude drift and only then comparing its position with ground truth. The figure shows the dependency of the drift on the amount of retained key-frames. We see that the gained performance decreases with the

Figure 2.3: Attitude drifts for different amount of retained key-frames. We observe failure modes of the visual pose estimation at around t=7 s and t=19 s. These failure modes are clearly visible as spikes. We make use of this in Section 3.1.2 to identify reliable visual pose measurements.

amount of retained key-frames. This is an expected result as the further away from the actual MAV pose a key-frame is, the less it affects the correction of a newly added key-frame. This fact is used in (Sibley et al. (2009)) for a relative bundle adjustment method. Due to the map corruptions at t=7 s and t=19 s, the original code still shows an offset after loop closing at t=30 s. Thus, we see the importance of keeping track of the drift states even in full SLAM frameworks.

At each position, we measured the median feature distance to the camera center. Since all features lie in the xy-plane of the ground truth frame, the ratio of that distance with the ground truth altitude yields a measure of the visual scale. We plot the scale drift in Fig. 2.5. The improvement gained by keeping more key-frames is clearly visible.

Fig. 2.6 shows the trajectories flown in 3D. As in the proceeding plots, we note that the original code is not at the correct position even after loop closing. This is due to the map corruptions during the loop.

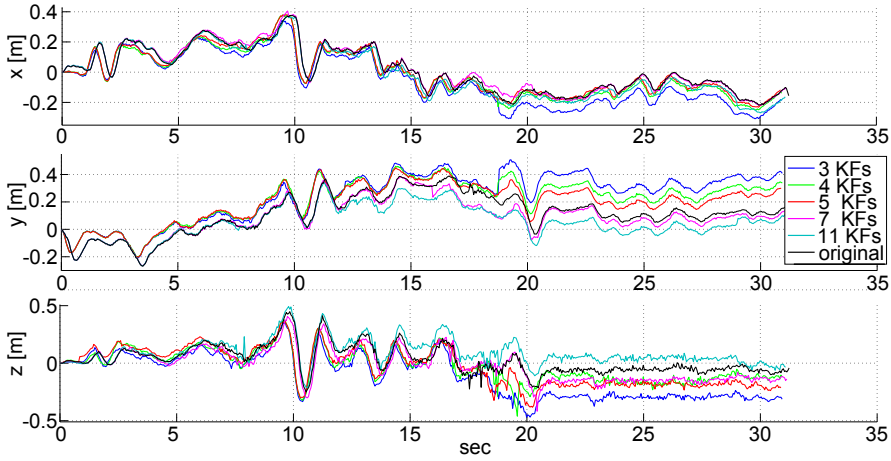


Figure 2.4: Position drifts for different amount of retained key-frames. Due to the failure modes and consequent map corruptions at second t=7 s and t=19 s the original code did not close the loop correctly at t=30 s. This shows the importance of keeping track of the drift states even in full SLAM systems.

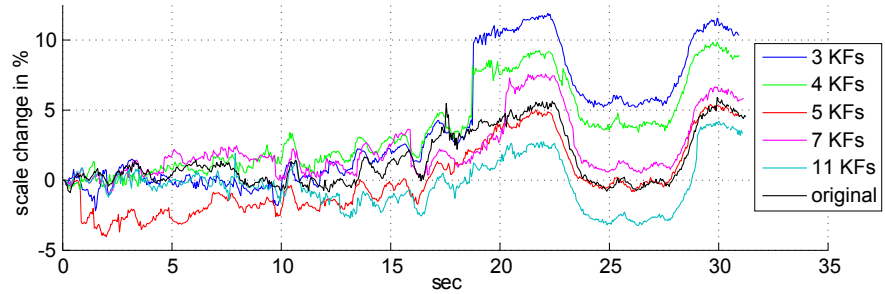


Figure 2.5: Scale drifts in percent for different amount of retained key-frames. The scale is calculated by the ratio of the median feature depth with the ground truth altitude. All features lie in the xy-plane of the ground truth frame.

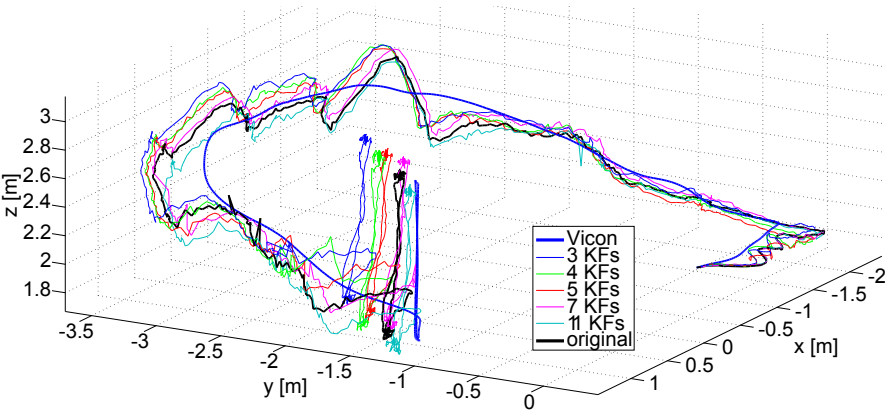
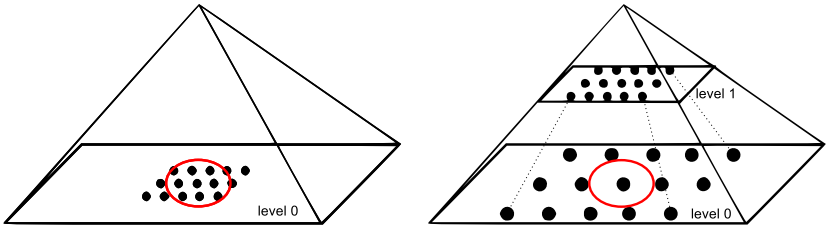


Figure 2.6: 3D trajectories of the different runs each time retaining a different amount of key-frame. The more key-frames we keep the less is the incremental improvement on performance. This is in accordance with (Sibley et al. (2009))

Improved Feature Handling for More Robust Maps

This improvement aims at selecting high quality features for robust map generation and camera pose tracking. Outdoors, self-similarity of the environment is an omnipresent issue – e.g. the asphalt in urban areas or the grass in rural areas. Furthermore, self-similar structure in outdoor environments is often high frequent (grass, asphalt). The higher the resolution of the camera, the more are these high frequent self-similar structures captured. Conversely, low resolution cameras blur these structures very similar to a low pass filter. A similar effect have the down-sampled pyramidal versions of a high original image which are stored in the map's key-frames. Thus, features extracted at higher pyramidal image levels are more robust to high frequent scene self-similarity. Features in higher levels represent salient points in lower frequencies whereas features in lower pyramidal levels represent salient points in high frequencies. Fig. 2.7 depicts the situation schematically.

Low frequent self-similarity is already well handled by the algorithms motion model limiting the search radius for corresponding features (red circle in Fig. 2.7(b)). In contrast, the search radius contains several matching candidates on high frequent self-similar structure Fig. 2.7(a). A logic step is thus to only include features extracted in the higher pyramidal levels for the map building process. This directly avoids issues with high frequent self-similar structures. We verified our intuition by analyzing the feature quality per



(a) (b)

Figure 2.7: a) Features of high-frequent self-similar structures are extracted in the lowest pyramidal level since most details are preseved in this level. The search radius calculated from the algorithm's motion model (red circle) covers several potential matches which leads to a wrong data-association. b) Down-sampling the original image (i.e. higher pyramidal levels) acts as a low pass filter. Only features of low-frequent self-similar structures are extracted and the motion model can disambiguate well beween the different features.

pyramidal level. As a metric, we use the percentage of outliers in the newly added features per key-frame. The outliers are detected in a local bundle adjustment step after a new key-frame with features is inserted. The graphs in Fig. 2.8 depict the different percentages and Fig. 2.8(c) shows a typical camera image of our outdoor data-set manifesting the issue of high-frequent self-similarity. In average, more than 45% of the newly triangulated features are marked as outliers in the consequent bundle adjustment process. Fig. 2.8(a) shows, that the majority of the outliers (96.04%) are wrongly triangulated features in the lowest pyramidal level. In fact, only about every third feature triangulated in the lowest pyramidal level is apparently an inlier whereas higher pyramidal levels show drastic improvements on the inlier/outlier ratio (Fig. 2.8(b)). Wrong triangulation is a direct result of false positives in the matching process. Since the algorithm's motion model is not very precise, the feature search region is larger than the spatial image periodicity in the lowest pyramidal level. This inevitably leads to wrong matches. How many of the lower pyramidal levels should be discarded for the map building process depends on the resolution of the original camera image and the frequency of the self-similarity in the mission area. We suggest to analyze the inlier/outlier ratio as we showed in Fig. 2.8(b) in order to define the cut-off pyramidal level.

We take advantage of the decoupled mapping and tracking processes of the framework and still track map features expected in the lowest pyramidal level

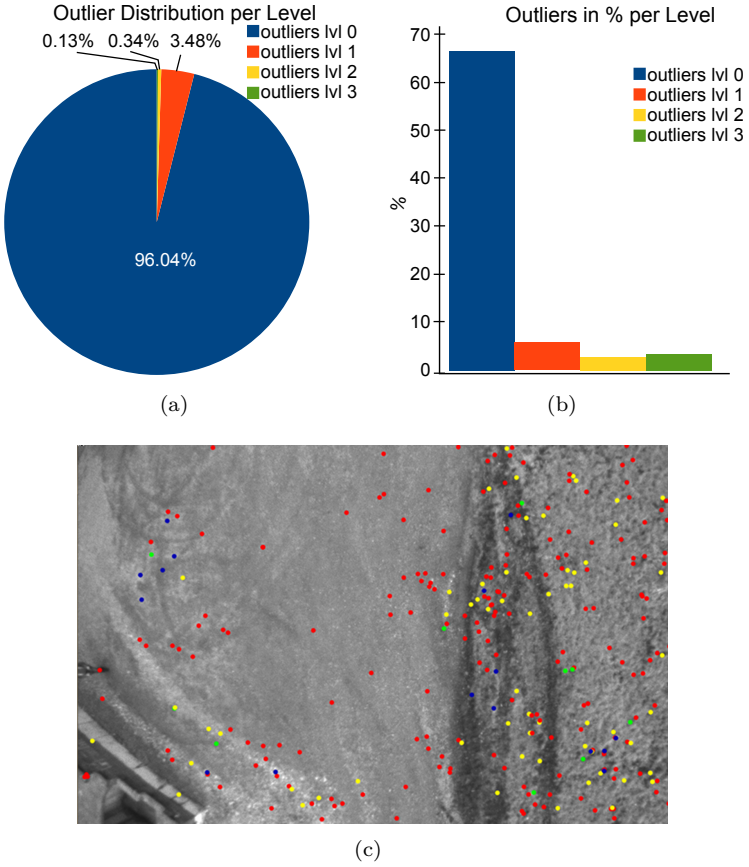


Figure 2.8: a) In total, about 45% of the features triangulated during a new key-frame additions are marked as outliers in the consequent bundle adjustment step. 96.04% of these outliers are in the lowest pyramidal level. b) We normalize the ouliers per pyramidal level with respect to the amount of triangulated features per level. We see that, even relatively, the features in the lowest level are the least reliable ones. c) a typical scene of an outdoor data-set (lvl0-lvl3: red, yellow, green, blue dots). We note the high-frequent self-similarity which causes triangulation in the lowest pyramidal level to fail. Lower frequent self-similarity is handled well by the algorithms motion model and the resulting restricted search areas. Thus, features in higher pyramidal levels are more robust.

of the current camera frame. This improves tracking quality when moving away from a given feature and makes it possible to navigate over both grass and asphalt. For wide angle cameras, it has also the advantage of being able to track peripheral features. We show these two situations schematically in Fig. 2.9. Fig. 2.10 shows in the top left part of a real outdoor image. The red dots represent previously added higher pyramidal level map features tracked at the lowest level in the current camera frame. The MAV flies from top left to bottom right, hence we already mapped the top left area and explore towards the bottom right new features. We show these two situations schematically in Fig. 2.9.

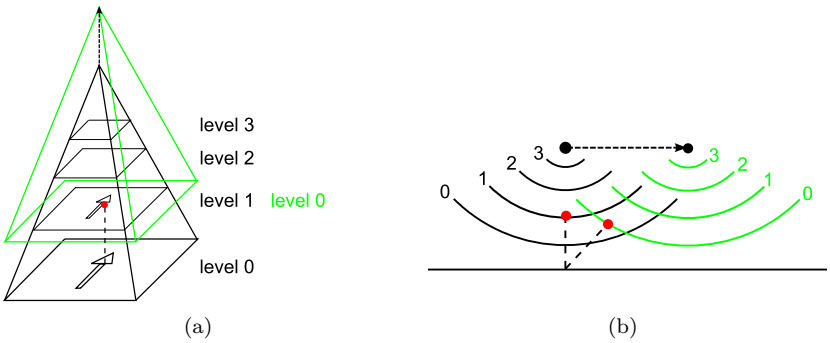


Figure 2.9: Two schematic situations where tracking features in the lowest pyramidal level is crucial. The black position represents the key-frame where the feature (red dot) got added to the map. The green position represents the current camera frame in which we try to find the previously mapped feature. a) Situation when moving away from a previously mapped feature: the resolution of the lowest pyramidal level in the more distant current frame becomes equivalent to the first pyramidal level of the key-frame. Hence the feature will be tracked in the lowest pyramidal level. b) Similarly, for wide angle lenses, the pripheral features are more distant than centric features. Hence, peripheral features usually will be tracked on lower pyramidal levels than centric ones.

We note that a wrong data association in the tracking part is less sensitive for the overall algorithm. First, wrong data associations are averaged over all features used to determine the camera pose. Second, a wrong estimate is temporary and may lead to a short jitter, but nothing is stored as a constant, wrong point in the map.

Since this vision algorithm is key-frame-based, it has high measurement rates when tracking. However, at key-frame creation, the frame-rate drops

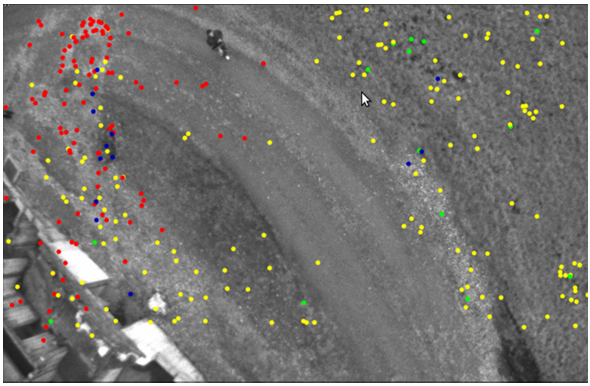


Figure 2.10: The map only contains features triangulated in higher pyramidal levels. When using cameras with a wide angle of view or when moving away from features, higher pyramidal level map points may be found in the lowest pyramidal level of the current camera frame. In the image, we use a 120° fisheye lens. Features near the rim are naturally further away. Nevertheless they get tracked correctly in the lowest level (red dots).

remarkably on on-board single core architectures. From Fig. 2.11(a) (feature distribution before outlier removal) and Fig. 2.11(b) (feature distribution after outlier removal) we know that the lowest pyramidal level not only yields most outliers in percentage, but also yields more features compared to higher pyramidal levels. In Fig. 2.11(c), we show the timings for a key-frame creation and see that the most time demanding parts are on a per-feature base: Shi-Tomasi score computation, FAST non-max suppression, and the addition of new features to the map. As we have seen, the lowest pyramidal level yields the least reliable information but most features. Neglecting this level reduces drastically the amount of features to be treated while creating a new key-frame. Thus, it directly reduces the computational cost and augments the map quality.

This results to great speed-ups with key-frame creation running at 13Hz (in contrast to the 7Hz of the original PTAM) and normal tracking rates of around 20 Hz on an on-board Atom single core computer 1.6 GHz. These timings are calculated using a maximum of 5 key-frames for the map representation in both the original and modified version.

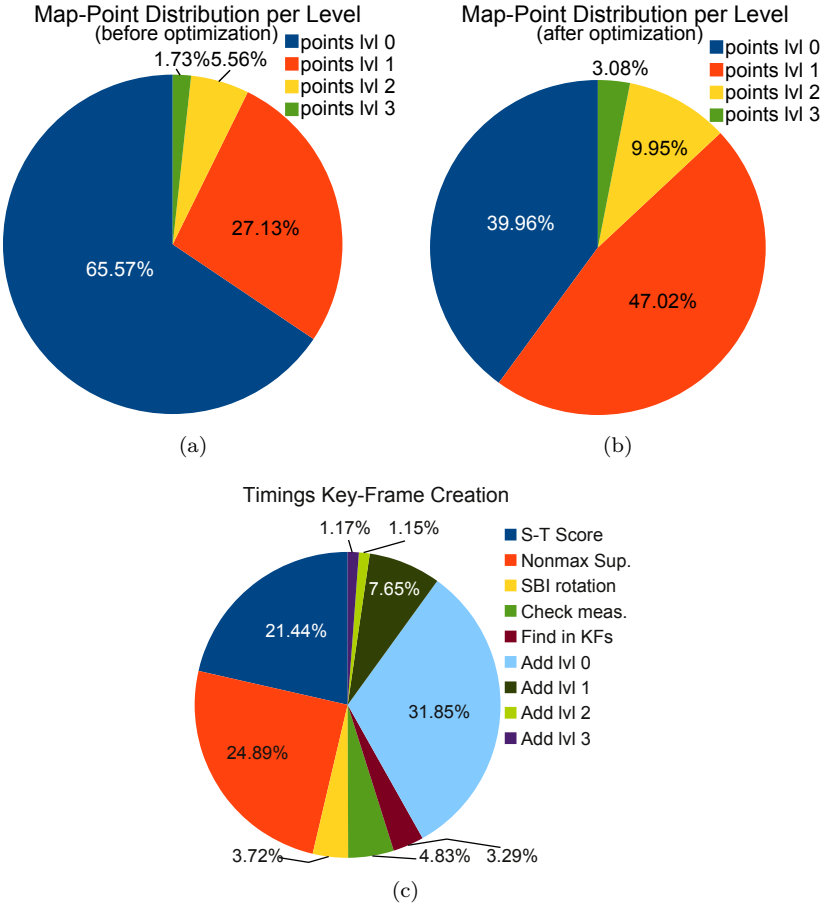


Figure 2.11: a) Feature distribution per level on key-frame creation before applying a bundle adjustment step. Roughly $\frac{2}{3}$ are features in the lowest pyramidal level. b) Feature distribution after the bundle adjustment step. We see that in the lowest pyramidal level, not only there are most features, but also most are rejected as outliers. c) shows the time spent on the different tasks for creating a key-frame. All major tasks are on a per-feature base. This means we can directly save computation time when reducing the number of features. ((c) from (Lynen (2011)))

Re-Initialization After Failure Mode

For automatic initialization we ensure that the baseline is sufficiently large by calculating the rotation-compensated median pixel disparity. For rotation compensation we use efficient second-order minimization techniques (ESM) (Malis (2004)) in order to keep PTAM independent of IMU readings. This keeps our approach modular to other 6DoF pose estimators. For re-initializations, we store the median scene depth and pose of the closest, last valid frame and propagate this information to the new initialized map. This way we minimize large jumps in scale and pose after re-initialization. In Fig. 2.12 we actively initiated two re-initialization steps while hovering. The first re-initialization at t=50 s barely affected the algorithm at all. At the second re-initialization step at t=72 s we notice a slight scale change occurred after successful re-initialization. This is represented in the drop of altitude with respect to ground truth and is due to the movement in the 2.5 seconds without any map.

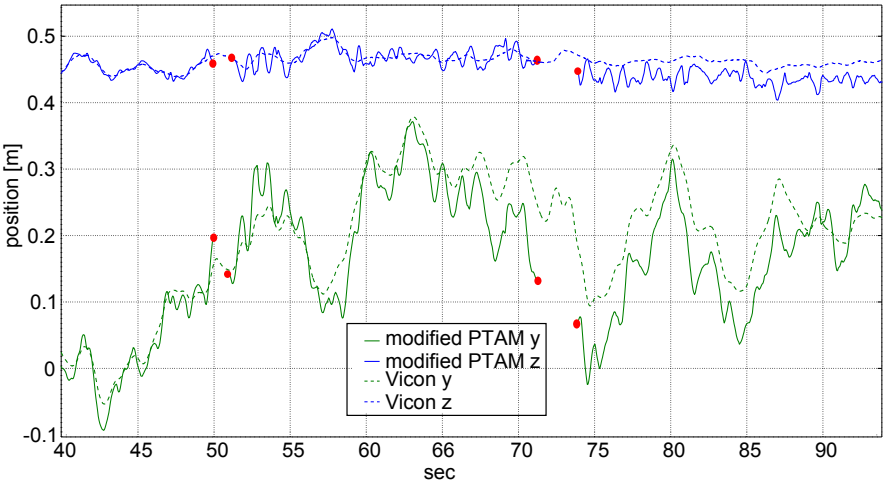


Figure 2.12: The graph shows 2 re-initialization steps at t=50 s and t=72 s. The gap between the red dots mark the initialization phase. The visual estimate is scaled and shifted to match the Vicon ground truth. We only applied this at the beginning and note that our scale and pose propagation methods upon re-initialization work well. A slight scale shift is observed after the second re-initialization: the estimated z-value is lower than before and lower than the ground truth.

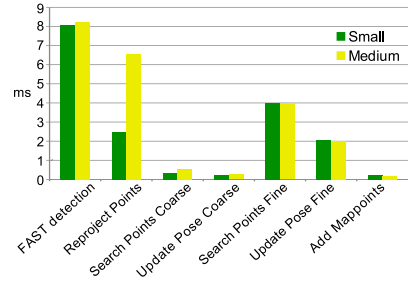
	Dense Map	Sparse Map
Original	10.4892 ms (std: 1.2188 ms)	8.01216 ms (std: 1.6215 ms)
Modified	11.0472 ms (std: 3.3256 ms)	1.3222 ms (std: 0.4153 ms)

Table 2.1: Tracking Timings for different map structures (Lynen (2011))

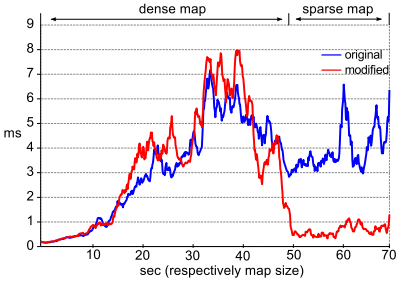
Inverted Index Structure for Map-point Filtering

This improvement aims at an intelligent selection of map-points in order to increase tracking speed in large and sparse maps. On each frame, the original PTAM projects the 3D points from the map into the current image according to the motion-model prior, which allows point-correspondences to be established for tracking. Since no filtering on point visibility is preceding this step, it scales linearly with the number of points in the map. Conversely, other steps performed on each camera frame for tracking (FAST detection, search the potentially visible map points in the image, update the camera pose, and add new features to the map) are independent of the map size. Fig. 2.13(a) lists the camera pose tracking timings for different map sizes.
We implemented an inverted index structure based on the grouping of map points inside key-frames which allows discarding large groups of map-points with low probability of being in the field-of-view (Lynen (2011)). The search for visible points is performed by re-projecting a small set of distinct map-points from every key-frame which permits inference on their visibility from the current key-frame. The total number of points which need evaluation by re-projection is thereby significantly reduced. This leads to a scaling of the system in linear order of the visible key-frames rather than in linear order with the overall number of points in the map.

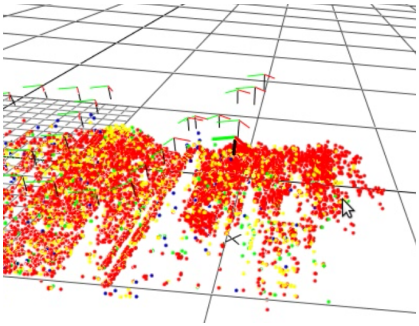
Fig. 2.13(b) shows the performance of our modification. Initially we moved the camera in loops. In this situation, many key-frames have to be considered in the current camera image and our modification performs worse than the original code because of the computational overhead of indexing the features. In the second part of the data-set, we moved on a straight line. In such situations we outperform the original framework clearly because of the selective feature re-projection. On large-scale maneuvers, the current camera frame often has to consider only a part of the map. Thus our approach yields a crucial performance gain in these situations. Fig. 2.13(c) and Fig. 2.13(d) show the situation where the overhead of our modification is dominant and where performance gain due to selective feature re-projection is dominant. Table 2.1 summarizes the results for the two different situations.



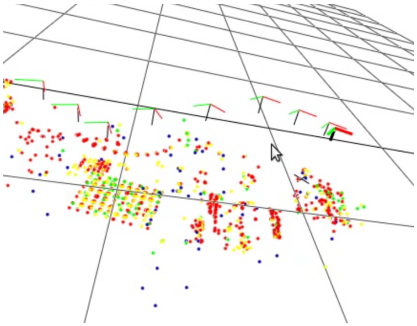
(a)



(b)



(c)



(d)

Figure 2.13: a) timings of the different steps performed on each camera frame for tracking. Only the re-projection of map points is dependent on the map size. Thus we aim at decrease its computational complexity. b) Comparison of timings between our modified code (red) and the original code (blue). If many key-frames are marked as visible (dense map), then the overhead of our modification is large and the original code outperforms our code. Navigating in large environments implies that a large part of the map is not visible (sparse map) then our modification clearly improve performance. The green dots mark the screen-shots of c) and d). c) depicts a typical situation of a dense map whereas d) depicts a sparse map. (Lynen (2011))

2.1.3 Feasibility Study

In order to test the validity of the visual pose estimate we apply the estimate to a linear optimal controller based on a cascaded structure and designed by means of a discrete LQG/LTR procedure. In the feasibility study we are interested in the visual pose estimate as such, thus we did not apply any sensor fusion with inertial readings. Also, we apply the visual scale manually. We discuss the complete sensor-fusion strategy for MAV state estimation and sensor (self-)calibration in Chapter 3.

To overcome the problem of map drift, we implement a simple correction step in our controller. This means, we read the inertial measurements in hovering mode and re-align the map according to the inertial frame. We apply this step every time after a predefined traveling distance Δd. It is meaningful to apply this re-alignment step independent of time since the drift is of spatial nature and not of a temporal one. These correction steps do not improve the consistency of the global map. However, they assure accurate local pose estimates making it possible to validate our vision based approach for MAV navigation. We published this study in (Bloesch et al. (2010)) and did further experiments outdoors with simplistic sensor fusion in (Achtelik et al. (2011b); Weiss et al. (2011b)).

Since the control part goes beyond the scope of this dissertation, in the following, we only list the achieved results of the feasibility study.

Proof of Concept

In Fig. 2.14 the flight path for a 60-seconds hovering can be seen. Because the mapping thread goes almost into sleep mode once a region has been explored, the computational power can be fully used for tracking and controlling the MAV. In addition, the map will not get corrupted in this state and the localization quality is guaranteed. Overall, the position error has an RMS value of 2.89 cm in x, 3.02 cm in y and 1.86 cm in z, which yields an absolute RMS error value of 4.61 cm (see Fig. 2.15).

To test the ability of way-point following we applied a set of step inputs and verified the controller performance. Fig. 2.16 shows that the MAV can robustly follow one way-point after the other.

The platform is also able to fly to desired set-points. For that, the path is split into way-points. The distance between them is chosen such that the helicopter can realign the orientation of the inertial coordinate frame at each way-point. With this setup, a slightly higher RMS value is obtained as in the hovering mode (see Fig. 2.17). We start with a map consisting of 7 key-frames. While expanding the map, the algorithm has processed more

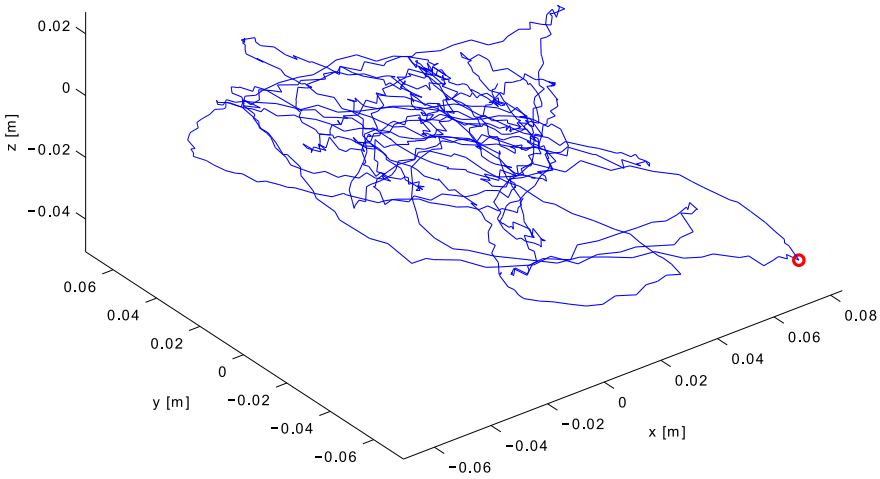


Figure 2.14: Position error while hovering during 60 seconds. The RMS value of the position error is 2.89 cm in x, 3.02 cm in y, and 1.86 cm in z. The maximum runaway has an absolute error value of 11.15 cm (marked with a red o). (Bloesch et al. (2010))

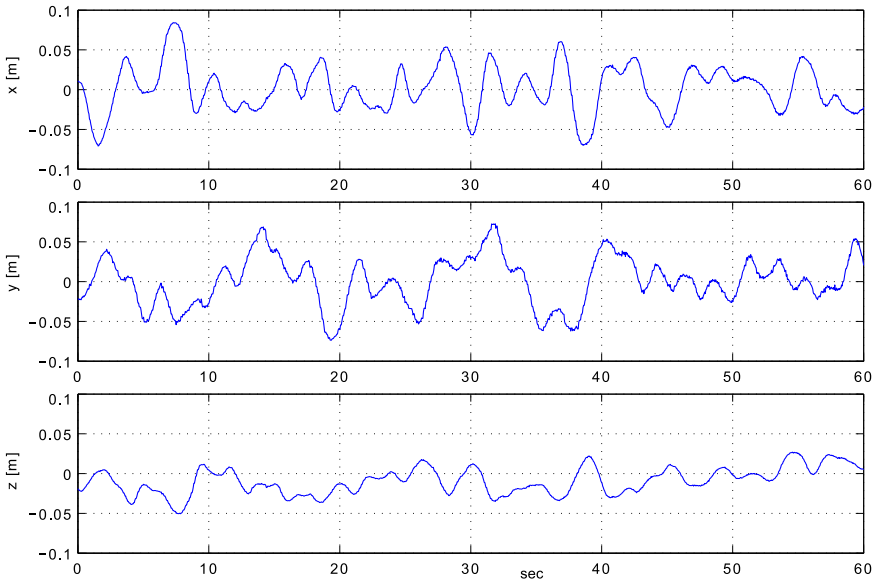


Figure 2.15: Position error in the x, y, and z positions. The value remains between $\pm$ 10 cm. The z position is more accurate than the x and y positions. (Bloesch et al. (2010))

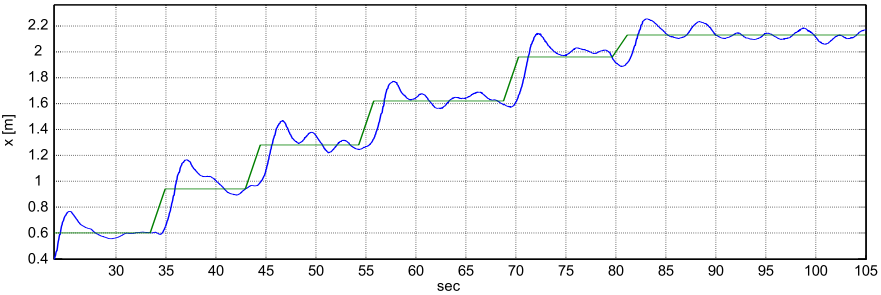


Figure 2.16: Several step inputs are applied one after the other to the real system. This is an intermediate step towards an ongoing way-point following. The plot shows that the system is able to follow robustly the applied sequence and is thus capable for following any trajectory. (Weiss et al. (2011b))

than 40 additional key-frames. At each way-point, the helicopter stabilizes itself and waits until the RMS value is small enough (10 cm in absolute error value) to realign the map. At the same time, this leaves some time for the algorithm to process new key-frames and expand the map. This shows the idea of having only a locally consistent map to control the helicopter. The realignment leads to a correct map representation in the close neighborhood. The map might not appear globally consistent but, nevertheless, the vehicle is well controlled at any position.

At some points, when the helicopter flies from way-point to way-point, a notable overshoot can be observed. This is due to the integrating action of the controller, which is necessary to correct steady state errors. It does introduce additional overshoot and reduces the robustness of the system.

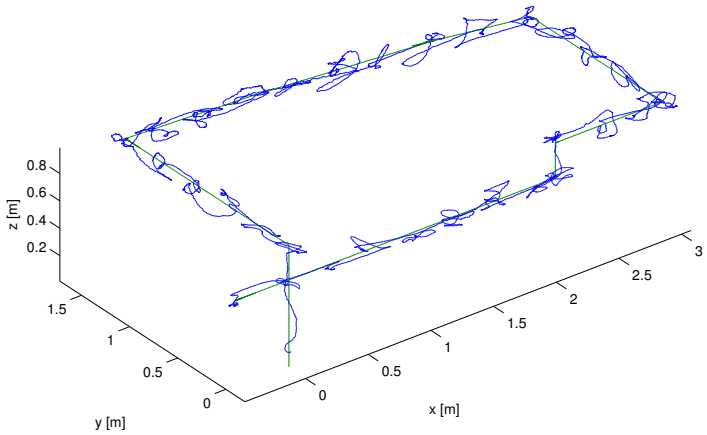


Figure 2.17: Path that the helicopter has flown. This does not represent ground truth. It is the pose estimation of the SLAM algorithm. However, the attitude of the helicopter can be observed while successfully flying a rectangular loop and landing on the ground. The RMS value of the position error is 9.95 cm in x, 7.48 cm in y and 4.23 cm in z. The path has a total length of a little bit more than 10 m in an area of 3.5x2x1 m^3. (Bloesch et al. (2010))

In Fig. 2.18 the map that was built during the flight, can be seen. It is composed of 52 key-frames and 4635 map points. It represents an approximate surface of 15 m^2. The micro helicopter will always localize itself correctly with respect to the neighboring local maps.

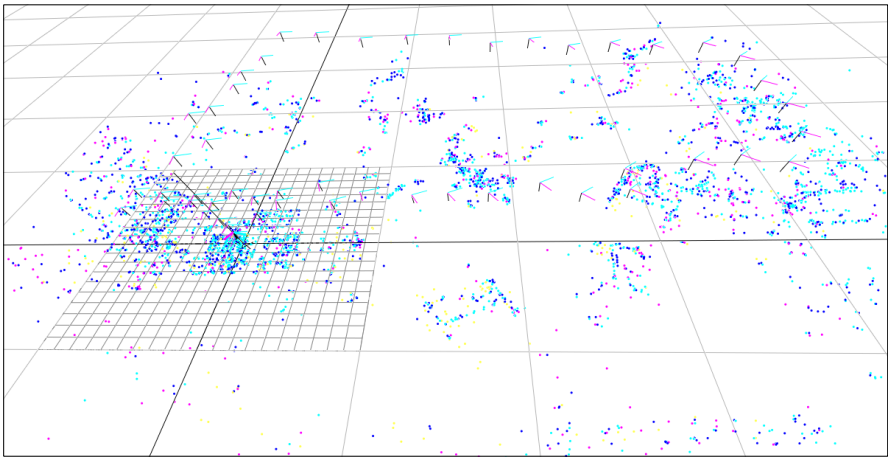


Figure 2.18: 3D view of the built map. It contains 4635 map points, observed in 52 different key-frames. (Bloesch et al. (2010))

The achieved results show that our platform can autonomously fly through a large unknown indoor environment with a high accuracy. The system is robust against external disturbances and can handle modeling errors. Outdoor tests confirm the controller's robustness, which was able to handle considerable side winds and turbulence.

2.2 Map Less Approach: The Camera as a Body-Velocity Sensor

In this section we focus on the proposed velocity-estimation module which converts the camera into a body-velocity sensor. This work has been presented in (Weiss et al. (2012b)). We focus here only on the vision part which is based on optical flow and the continuous 8-point algorithm (Ma et al. (2000)) augmented with IMU readings. This addition drastically reduces the dimensionality of the solution space. In contrast to the previously discussed map-based approach, this *inertial-optical flow* approach does not need any sort of map or feature history. In fact, two feature matches in two consecutive camera frames (i.e. optical flow) are sufficient for our method to recover the body velocity. Pose drifts and map losses are thus not an issue making it an ideal, complementary method to the aforementioned map-based approach. Using our scene depth normalization, we also circumvent the scale drift issue

to a large extent.

Employing optical flow in visual-inertial tracking is increasingly popular but existing systems either demonstrate results only in simulation (Bleser and Hendeby (2010); J. L. Center and Knuth (2011)), transmit images to a ground station for off-board processing (Hérissé et al. (2010); Schill et al. (2011)), or use special cameras only suitable for optical flow calculation (Zufferey (2005)). In this chapter, we demonstrate real-time inertial-optical flow based body-velocity recovery with a conventional camera running at 40 Hz while all processing is done on-board on an ATOM computer 1.6 GHz.

At this point, it is worth mentioning again that on an airborne vehicle, estimating a goal vector for control is not sufficient. Unlike ground vehicles, airborne vehicles cannot simply hold all actuators still to achieve zero velocity. Hence, for robust MAV control, a timely estimate of its actual state is a requirement – then, based on this estimate, goal vectors can be applied. As a result, we will focus on the prompt and consistent availability of the MAV velocity estimate, allowing goal vectors to be generated by simply moving set-points or holding a set-point for hover mode. We do not focus on the control itself but refer to our work in (Bloesch et al. (2010)) and (Achtelik et al. (2011b)).

Later, in Chapter 3.4, we focus on the semi-tightly coupling of our inertial-optical flow method with an EKF framework. There, we apply a non-linear observability analysis to prove the observability of the visual scale as well as the inter-sensor calibration between camera and IMU. This allows full sensor (self-)calibration and metric MAV pose estimation in real-time for accurately velocity control the aerial vehicle. The inertial-optical flow module is then applied both during initialization and as a fall-back solution at tracking failures (e.g. map loss) of a map-based module.

2.2.1 2D Continuous "8-Point" Algorithm

We assume $\vec{x}$ and $\vec{x'}$ to be the feature direction-vector and in the first and second camera frame respectively. Then, the discrete epipolar constraint is $\vec{x'}^T E\vec{x} = 0$ where the essential matrix $E(T, R)$ consists of the camera rotation R and translation T between the two camera frames. For two infinitesimal close camera frames, in continuous space, we can calculate continuous equivalents of T and R, the translational and angular velocities v and ω respectively. For each 3D point's coordinates $\mathbf{X}(t)$ the following holds:

$$\dot{\mathbf{X}}(t) = \lfloor\vec{\omega}(t)\rfloor\mathbf{X}(t) + \vec{v}(t) \tag{2.2}$$

Introducing the arbitrary scale factor λ yields $\lambda(t) * \vec{x}(t) = \mathbf{X}(t)$. We sub-

stitute $\vec{x}$ by $\vec{u}$ as the optical flow vector and obtain the following continuous epipolar constraint:

$$\vec{u}^T \lfloor \vec{v}(t) \rfloor \vec{x} + \vec{x}^T \lfloor \vec{\omega}(t) \rfloor \lfloor \vec{v}(t) \rfloor \vec{x} = 0 \tag{2.3}$$

with the skew symmetric notation of a vector cross product $\lfloor \vec{a} \rfloor \vec{b} = \vec{a} \times \vec{b}$. As mentioned in (Ma et al. (2000)), in contrast to the discrete version, solving for the continuous essential matrix yields a unique solution for $\vec{v}$ and $\vec{\omega}$ as the twisted-pair ambiguity is avoided. Moreover, the continuous approach handles zero-baseline situations well, avoiding the singularities of the discrete case.

The system in (2.3) requires 8 features with their corresponding optical flow vectors. From the discrete version, we know that the problem in fact only spans 5 dimensions (3 for each of rotation and translation and -1 for the unknown scale). Here, we can also incorporate the knowledge of angular velocities from an attached IMU, bearing in mind that the IMU needs to be time-synchronized with the camera (i.e. temporal synchronization, which is ensured in our setup). Furthermore, the spatial calibration between IMU and camera needs to be known – the latter is addressed in the next chapter (Section 3.4). This eliminates 3 more dimensions such that only 2 dimensions remain (i.e. the direction of the velocity). Measuring the angular velocities with the IMU allows to unrotate the optical flow and allows to set ω in (2.3) to zero. Then the 2 dimensions of the new problem are immediately visible given that the velocity can be arbitrarily scaled. The new problem can now be formulated as:

$$\begin{aligned} \vec{u}^T \lfloor \vec{v}(t) \rfloor \vec{x} &= 0 \text{ , or equivalently} \\ (\lfloor \vec{u}(t) \rfloor \vec{x})^T \vec{v} &= 0 \end{aligned} \tag{2.4}$$

using the properties of the triple product. Equation (2.4) suggests that the camera velocity $\vec{v}$ is orthogonal to the cross product of the optical flow $\vec{u}$ and the feature direction vector $\vec{x}$. In other words, $\vec{v}$ lies in the plane spanned by $\vec{u}$ and $\vec{x}$. Geometrically, the intersection of two such planes unambiguously defines the direction of $\vec{v}$ (Ma et al. (2000)). This is already sufficient since $\vec{v}$ can be arbitrarily scaled, thus we need at least two vectors $\vec{u}$ and $\vec{x}$ to define $\vec{v}$ up to scale. Fig. 2.19 depicts this schematically.

With the above, we have an arbitrarily scaled visual velocity by only observing at least 2 features and their current optical flow. Since any two features in a non-degenerate configuration yield this information, we do not need to store any feature history but instead solely depend on the last two camera frames.

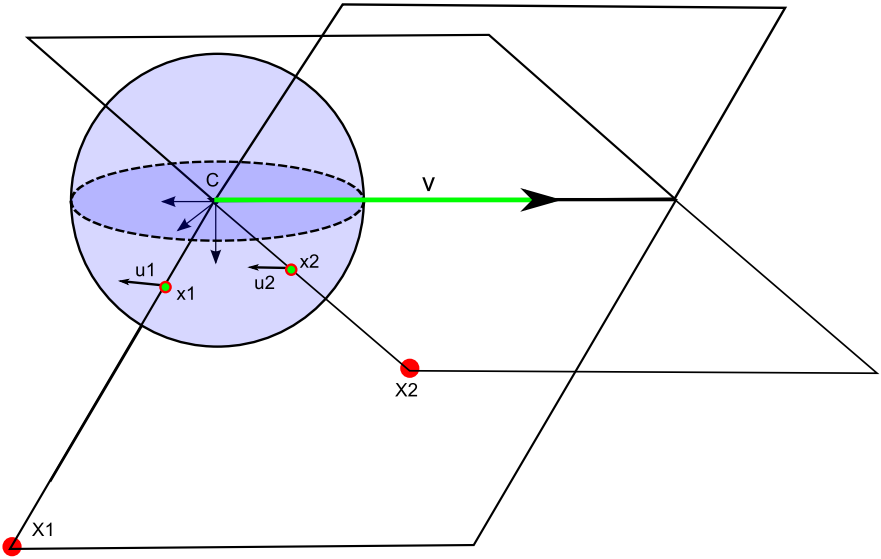


Figure 2.19: Graphical setup for the continuous epipolar constraint given IMU measurements for the rotational velocities. The constraint simplifies to (2.4) such that only 2DoF (i.e. the direction of the velocity vector $\vec{v}$) remain. The equation suggests that this direction is in the plane spanned by a feature direction vector $\vec{x}$ and its optical flow $\vec{u}$. In other words, the triple product of these 3 vectors vanishes. Using a second feature X_2 and intersecting this second plane with the first, uniquely defines the velocity direction vector $\vec{v}$. (Weiss et al. (2011b))

Recovering a Unifying But Arbitrary Scale Factor

So far, we recovered the camera velocity up to an arbitrary scale. Without loss of generality we can set $|\vec{v}| = 1$. We can then rewrite (2.2) as

$$\dot{\mathbf{X}}(t) = \lfloor\vec{\omega}(t)\rfloor\mathbf{X}(t) + \eta\vec{v}(t) \ , \tag{2.5}$$

with η being an arbitrary scale factor for the recovered unit norm velocity. We still un-rotated the optical flow and have thus $\omega = 0$. Introducing the feature scales λ_i per feature i lets us rewrite (2.5) as

$$\dot{\lambda}_i(t)\vec{x}_i(t) + \lambda_i(t)\dot{\vec{x}}_i(t) = \eta\vec{v}(t) \ . \tag{2.6}$$

We can stack all scale factors λ_i, their temporal derivatives $\dot{\lambda}_i$ and the common scale factor η for the velocity $\vec{v}$ into a single vector

$$\vec{\lambda} = [\lambda_1, \ldots, \lambda_n, \dot{\lambda_1}, \ldots, \dot{\lambda_n}, \eta] \tag{2.7}$$

Since (2.6) is linear in $\vec{\lambda}$ we can write it as

$$M\vec{\lambda} = 0 \ , \tag{2.8}$$

with M depending on the unit scaled velocity $\vec{v}$, the feature direction vectors x_i and their optical flow $\dot{x}_i$. Note that (2.8) unifies the scale factors to be consistent with each other in this particular camera reading. That is, λ_i of feature i is larger than λ_j of feature j if feature i is further away than feature j. We can put this formally as $\vec{\Lambda} = L\vec{\lambda}$ with $\vec{\Lambda}$ being the metric distances to the features and L the arbitrary but unifying scale factor by which our visual estimate needs to be scaled to receive metric values. For the map less approach we have in summary the following different scale factors: λ_i scales the unit direction vector $\vec{x}_i$ to some arbitrary vector $\mathbf{X_i}$ such that its length is consistent with the scene. The factor L scales then the whole scene to metric units - this factor is conceptually the same as the visual scale factor in the map based approach in Section 2.1.

In (Ma et al. (2000)) the authors use the solution of (2.8) to find a globally and temporally consistent scale factor for subsequent camera frames (i.e. temporal scale propagation). More precisely, $\vec{\lambda} = \vec{\lambda}(t)$ is as a function of time. Since $\vec{\lambda}(t)$ is determined up to an arbitrary scale at each point in time, $\rho(t)\vec{\lambda}(t)$ is also a valid solution for any positive function $\rho(t)$. However, the ratio

$$r(t) = \dot{\lambda}(t)/\lambda(t) \tag{2.9}$$

is independent of $\rho(t)$. This is a differential equation for $\lambda(t)$ with the solution

$$\lambda(t) = \exp(y(t)) \tag{2.10}$$

with $y(0) = y_0 = \ln(\lambda(0))$ and $\dot{y}(t) = r(t)$. This means, that the scale estimates at different points in time are all relative to the scale estimate at $t = 0$. In practice, this propagation suffers from integration errors because of discretization.

While the discussed appoach for temporal scale propagation is certainly meaningful for general environments and camera motions, we can exploit some soft constraints for a setup of a down-looking camera mounted on a micro helicopter for another scale propagation approach. For slow paced helicopter movements, we can assume the roll and pitch angles to be small. Furthermore, while looking down, the environment does not change abruptly from one frame to the other. We keep these assumptions in mind and have a closer look at the optical flow nature and definition.

It is important to note that with the same feature direction vectors x_i and optical flow $\dot{x}_i$, (2.4) and (2.8) will yield the same η and $\vec{v}$. From the definition of optical flow, the same readings for x_i and $\dot{x}_i$ can only occur if the ratio between the (metric) velocity and distance to the scene is constant. This is the case when the camera moves slowly close to the scene or fast far away from the scene. In the following we analyze the situation of an equi-scene-distant scenario. We assume to travel at velocity ${}^1\vec{v}$ always keeping the same distance to the scene (i.e. travelling parallel to the xy-plane if the scene lies on the xy-plane). Measuring $\vec{x}$ and $\dot{\vec{x}}$ will yield from (2.4) a unit direction vector $\vec{v}$ in the direction of ${}^1\vec{v}$. From (2.8) we will obtain ${}^1\vec{\lambda}$. If we travel double the velocity $2\ {}^1v = {}^2v$ in the same direction we observe the following: $\vec{v}$ from (2.4) remains unchanged since we travel in the same direction. We can apply the intercept theorem and obtain the relationship

$$\frac{\dot{x}_i}{x_i} = \frac{\dot{\lambda}_i}{\lambda_i} \tag{2.11}$$

Furthermore we know that the feature direction x_i remains unchanged but the optical flow changes according to $2\ {}^1\dot{x}_i = {}^2\dot{x}_i$ (we start at the same position but just travel twice the velocity). This means that

$$2\frac{{}^1\dot{\lambda}_i}{{}^1\lambda_i} = \frac{{}^2\dot{\lambda}_i}{{}^2\lambda_i} \tag{2.12}$$

(2.8) finds a consistent solution up to one unifying scale factor. If we set $2^1\lambda = {}^2\lambda$, (2.12)is reduced to $2^1\dot{\lambda} = {}^2\dot{\lambda}$ and evidently leads from (2.6)

to ${}^2\eta = {}^2\eta$. However, from (2.6), both vectors ${}^1\vec{\lambda}$ and ${}^2\vec{\lambda}$ are normalized to ${}^1\vec{\lambda}_N$ and ${}^2\vec{\lambda}_N$ with unit length. Thus ${}^2\eta_N \neq {}^2\eta_N$. If we re-normalize according to the scene depth

$$ {}^k\vec{\lambda}_{rN} = \frac{{}^k\vec{\lambda}_N}{\frac{1}{n}\sum_{i=1}^{n} {}^k\lambda_{Ni}^2} \tag{2.13} $$

then ${}^2\eta_{rN} = {}^2\eta_{rN}$ holds. Using this scene depth normalization and the conditions that the distance to the scene remains constant and that the body-velocity varies, η varies but the metric scale factor L remains unchanged. Conversely, if the body-velocity remains constant and the scene depth varies, the metric scale factor L changes but η remains unchanged.

Thus, L may be considered as an average indicator for the scene depth (or vehicle's height when using a down-looking camera) which changes as slow as the scene depth changes and which could be used for position drift compensation. However, this average depends on the current scene structure which we do not use in this sense. In our framework, we make use of the fact that the metric scale factor L does not change as long as we keep the distance to the scene constant. This means, it is a spatially and not temporally drifting factor. We will make use of this in Chapter 3.4 in order to eliminate the need of temporal scale propagation in the vision framework. Instead, this propagation will be handled by an underlying EKF framework including the IMU readings. As for now, we are only aiming at the scaled body-velocity $v_B = \eta\vec{v}$.

2.2.2 Performance of the Inertial-Optical Flow Framework

In this section, we evaluate our inertial-optical flow approach in simulation and in real experiments. In simulation we set a focus on both the effect of propagating the visual scale within the visual framework and of solely relying on our scene-depth normalization technique without explicit scale propagation. In the practical tests, we are particularly interested in the quality of the signal and the timings on an on-board platform. We emphasize that controlling the vehicle using only the vision input as we showed it in the map-based approach is not an option here. The difficulty in manually determining the true metric scale factor of the visually estimated body-velocity prevents testing this algorithm on real and agile platforms without proper sensor fusion. We discuss such results in 4.2.7 after developing the underlying fusion theory.

Simulations

We generated a 3D point cloud which is used as features. The simulated camera model implies pixel noise of 0.5 pixel. The trajectory is performed in all 6DoF, exciting the three position axis and the three rotational axis and around a starting set-point. In Fig. 2.20 we show the estimated body-velocity and the error with respect to ground truth. When we apply temporal scale propagation within the algorithm, we initially see an increased performance with respect to the approach where we solely apply scene-depth normalization. However, the accumulated error in propagating the scale leads to increasingly inferior performance. Conversely, when we only apply our proposed scene-depth normalization, the velocity suffers from more noise because of the lack of the smoothing in the scale propagation step. Nevertheless, the estimate is equally erroneous over time and around a certain scene depth. Over time, this outperforms the approach using scale estimation in the visual framework.

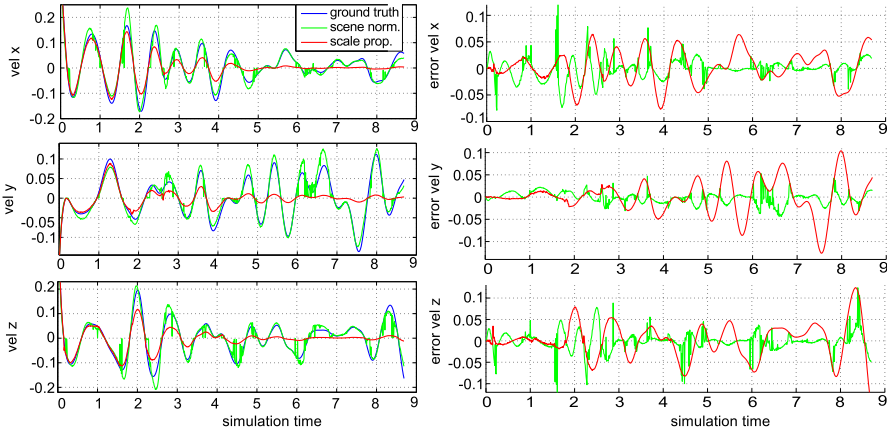


Figure 2.20: Performance of the inertial-optical flow based body-velocity recovery. The ground truth is visualized in blue. In red we plot the result when using temporal scale propagation within the visual framework. Because of the accumulated errors, the scale is prone to drift and affects the body-velocity estimate. Conversely, in green we plot the result of solely using scene depth normalization. It shows increased noise because of the lack of temporal smoothing, but proves to be consistent over time (per definition). The left plots show the actual velocity whereas the right plots show the corresponding errors with respect to ground truth.

In Fig. 2.21 we plot the evolution of the visual scale. We clearly see the scale drift in the scale propagation approach, whereas the scale of the

other approach remains (per definition) around a set-point at a cost of non-negligible outliers. We added the current scene distance to the plot to see their correlation. Using scene normalization only, the scale changes according to the scene distance (i.e. according to the MAV height when using a down looking camera). This is even better visualized in Fig. 2.21(b) where we sorted the scene depth in Fig. 2.21(a) in ascending order and plotted it together with the corresponding median-filtered scale. The correlation is evident.

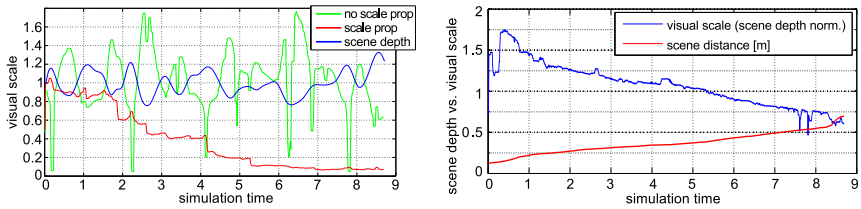


Figure 2.21: a) shows the scale evolution of the two approaches. In red, we plot the scale from the approach where we perform temporal scale propagation within the vision framework. In green, we plot the scale from the approach without explicit scale propagation. Red marks the scene distance. It shows clearly the correlation between the scene depth and the scale obtained by scene depth normalization. Also the noise is increased with respect to to the other approach. b) shows the sorted scene depth in red and the corresponding scale obtained by scene depth normalization.

Real Experiments

For the experiments on a real MAV, we used a hexacopter platform provided by Ascending Technologies[4]. The platform is equipped with an IMU and a WVGA monochrome camera with global shutter. The camera frame-rate was set to 20Hz whereas the IMU yields measurements at 1kHz. As ground truth data, we use a Vicon system with millimeter and sub-degree accuracy. We take the temporal derivative of the Vicon position for ground truth velocity.

For the first experiment, we moved the MAV hand-held about 0.5 m above the ground in all Cartesian directions. In Fig. 2.22 the estimated velocity vs. ground truth velocity is plotted. Evidently, the velocity is well estimated despite some obvious outlier-updates from the visual velocity measurement (e.g. after sec.44 in x, sec.56.5 in y, sec.74 in z).

In the second experiment we tested our approach on the flying vehicle. Fig. 2.23 shows the estimated body-velocity versus ground truth data. The

[4] www.asctec.de

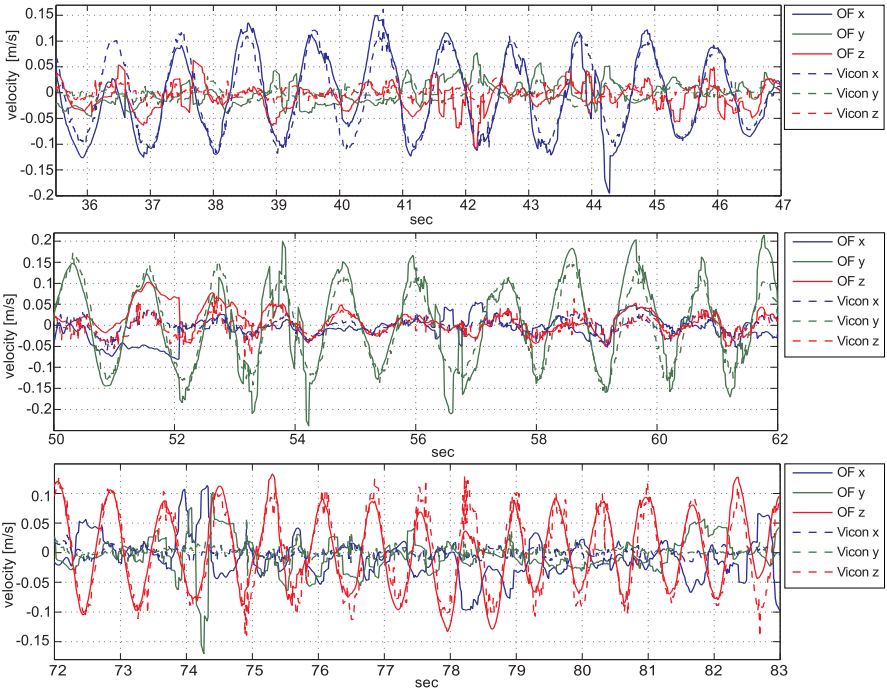


Figure 2.22: Estimated MAV velocity in x (blue),y (green),z (red) directions for a handheld MAV. The movements were made separately in x (top), y (middle) and z (bottom). Bold lines correspond to Vicon ground truth (noise arises from the position derivative of the Vicon data) and thin lines are the filter estimates. Notice that at points, the visual reading is corrupted, imposing a wrong update on the filter. Nevertheless, the estimates are robust with a RMS of [0.028, 0.035, 0.025] m/s in x,y and z, respectively. (Weiss et al. (2012b))

estimated body-velocity is manually scaled initially for better comparison with ground truth. This plot shows good performance of the algorithm even though considerable noise has to be taken into account when using this estimate for MAV velocity control. For stable MAV velocity control, not only sophisticated sensor fusion techniques have to be applied because of the noisy signal, but also because the inter-sensor calibration between IMU and camera as well as the sensor calibration of the IMU (i.e. biases) have to be sufficiently accurate. The high accuracy is needed to un-rotate correctly the optical flow readings of the camera.

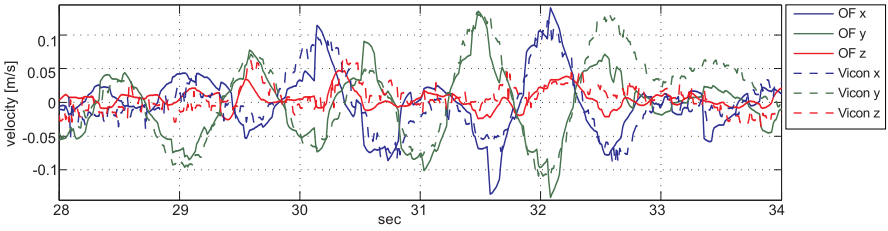


Figure 2.23: Estimated MAV velocity on an on-board flying MAV (thin lines) compared to ground truth (bold lines). Even though the estimate is noisy, it follows the excitations of the MAV according to the ground truth data. Nevertheless, sophisticated sensor fusion techniques discussed in the next chapter have to be applied for stable MAV velocity control. (Weiss et al. (2012b))

The timings for the vision algorithm are listed in Fig. 2.24. Note that, on average, our inertial-optical flow approach can run at just under 40 Hz on an on-board Atom computer 1.6 GHz. We divided the algorithm in 3 parts: a) *Feature management* ensures enough features equally spread in the image – in the case of bad feature readings, this algorithm takes longer to define suitable features. b) *Feature extraction* and matching establishes correspondences in consecutive frames. c) *Visual velocity calculation* using sparse SVD methods to solve (2.4) and (2.6). The big difference in timings in Fig. 2.24 are due to both our sparse matrix implementation and our feature prediction methods for fast feature matching.

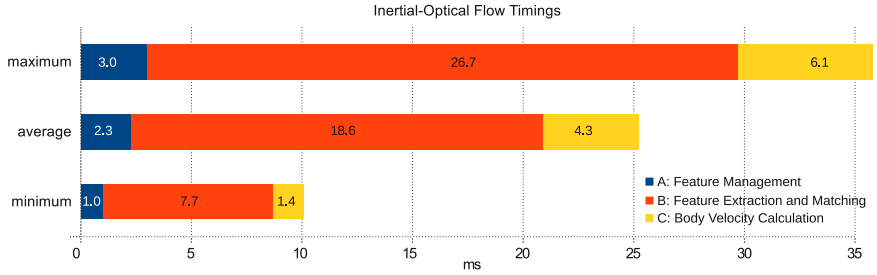


Figure 2.24: Timings of our inertial-optical flow to recover the visual velocity on an ATOM 1.6GHz. A corresponds to the feature management, B is the feature extraction and matching and C is the calculation of (2.4) and (2.6). The large difference between min and max timings arise from our feature prediction method for fast matching and the application of sparse matrix calculation. (Weiss et al. (2012b))

2.3 Conclusion

In this chapter introduced the camera as a viable *real-time* and *on-board* sensor for computationally constrained micro helicopters. We achieved both, a 6DoF scaled pose estimation up to 20 Hz and a 3DoF scaled body-velocity estimation at 40 Hz (mean value) on an on-board single core ATOM 1.6 GHz processor board only weighting 150 grams. We introduced the notion of visual scale and scale-drift. Also, we discussed the issue of visual position-drift and visual attitude-drift and their understanding with respect to a world fixed reference frame. This will be of importance in the next chapter, where we will discuss methods to mitigate such issues.

2.3.1 Camera as a 6DoF Pose Sensor

Concerning a visual pose estimation, we presented modifications to a state-of-the-art visual pose estimator in order to render the camera into a real-time 6DoF scaled pose estimator on a micro helicopter. Constraining the amount of key-frames used for the map drastically increased speed (mainly by eliminating the cost of a global bundle adjustment). Abandoning the lowest pyramidal features in the map increased performance on outdoor high-frequent self-similar structures and increased key-frame creation speed. In the original code, slow key-frame creation caused a significant performance drop on single core architectures leading to critical situations for the flying MAV.

Adding a re-initialization routine ensured continuous operation even in case of a map loss. We analyzed the effect of our modifications in particular on drifts but also on increasing robustness in self similar scenes and on timings.

In a feasibility study we then presented a vision based MAV control approach. The pose was estimated by means of the visual algorithm. This estimate was used to stabilize the position of the vehicle. Based on a control input transformation and on the linear LQG/LTR procedure, a controller was designed. The resulting platform successfully managed to hover and follow desired set-points within an indoor laboratory. For this task, it does not need any prior information on the environment. After the initialization, a map of the surroundings was built incrementally, wherein the MAV was able to localize itself. The vehicle can control its position up to a few centimeters of error (RMS around 2-4 cm). We showed that this is possible by only having a local consistent map, making the controller largely independent of map drift. Using the proposed strategy, our helicopter is able to perform all basic flight maneuvers such as autonomous take-off, set-point following, and landing. Our feasibility study proves the viability of controlling a micro helicopter based on a visual pose estimate.

2.3.2 Camera as a 3DoF Body-Velocity Sensor

We proposed an inertial-optical flow approach capable of estimating the camera body-velocity such that an MAV could be controlled in velocity. Unlike state-of-the-art on-board approaches, we used a conventional camera while still processing all tasks on-board. The same camera can be used for the previously discusses map-based approach as well as for further tasks like obstacle avoidance, semantic perception and user feedback. This limits the amount of different sensors needed on the platform and, thus, directly reduces the overall payload.

We exploit inertial measurements in angular velocities to reduce the dimensionality of the state-of-the-art continuous 8-point approach. This led to a two-dimensional problem which can be solved efficiently in real-time on-board computationally constrained platforms. We focus on not relying on any feature history or map. Our approach only uses matched features in two consecutive camera frames to estimate a scaled 3D body-velocity vector. This renders the approach highly robust to failures and may thus be used to bridge failure modes of the previously presented map-based approach.

We analyzed a strategy to propagate the visual scale factor within the visual framework and showed the issue of a drifting scale factor due to accumulated error. This essentially leads to similar issues as on map-losses in

the map-based approach. Our second, and preferred, strategy is based on a normalization of the scene depth after each calculation step. This leads to a time independent and constant scale factor while maneuvering in equidistant planes with respect to the scene. The scale changes according to the change in scene depth. This spatial dependent and temporal independent scale factor is motivation for complex sensor fusion in the next chapter. Furthermore, we presented real-time performance of our approach on a real platform. We achieved an average body-velocity recovery rate of 40 Hz on an single core ATOM 1.6 GHz processor board.

Chapter 3

Modular Sensor Fusion: State Estimation and Sensor Self-Calibration

In the previous chapter we learned two approaches to render a single camera into a valid on-board and real-time motion sensor for either the scaled camera pose or camera body-velocity. Whereas state-of-the-art uses off-board processing and/or artificial landmarks for vision based micro helicopter pose control, we showed such control on-board, in real-time and in a previously unknown environment (Bloesch et al. (2010)). Furthermore, whereas state-of-the-art uses off-board processing and/or specific (1D-)cameras for (reactive) velocity control, we proposed an on-board and real-time inertial-optical flow based body-velocity recovery method with a conventional WVGA camera.

In this chapter, we discuss proper sensor fusion in order to estimate the micro helicopter's pose *and* the (inter-)sensor calibration of the full sensor suite. In a first step, we focus in particular on our core setup consisting of a single camera and an IMU as only sensors for locally stable MAV navigation. In a second step, we add further sensors for drift compensation. State-of-the-art mainly focuses either on the calibration of the sensors *or* on the pose estimation assuming known calibration. Furthermore, most approaches are poorly scalable and can thus barely be used for motion estimation in large environments. In our work, we aim at a *combined* (pose estimation and calibration) approach while running *on-board* and in *real-time* and at the same time suitable for *large environments*. The approach is based on an Extended Kalman Filter (EKF) sensor-fusion approach. The estimation of

the helicopter pose will allow us to apply simple control methods for navigation whereas the (inter-)sensor calibration ensures long-term consistency for navigation in large environments.

A carefully selected sensor suite can be used to increase the autonomy of a MAV and hence improve robustness of navigation. However, each sensor adds to the payload. As a rule of thumb, every 10 grams require 1 W of motor power in hover mode for a small helicopter. Our core-sensor setup consisting of a camera-IMU setup weights about 20 grams and provides locally sufficient stability for real-time, on-board control of the MAV without the need for artificial landmarks.

As in any multi-sensor system, the (inter-)sensor calibration is crucial to the robustness of our estimation processes. While we assume the intrinsic camera parameters to be known and fixed, the inter-sensor calibration parameters describing the 6DoF pose between the IMU and the camera and additional sensors are unknown. There exist various methods in the literature to calibrate these unknowns. Particular attention has been paid to calibrate the IMU and camera calibration parameters (Kelly and Sukhatme (2011); Mirzaei and Roumeliotis (2008)). However, these approaches usually address off-line calibration exhibiting complexity of at least $O(M^2)$ for M features observed by the camera. In this work, we aim at a *power-on-and-go* system which calibrates itself while flying. Thus, computationally complex methods are unsuitable. Instead, a non-linear observability analysis reveals that we can decouple our vision algorithm by treating the arbitrarily scaled 3D camera velocity or 6Dof pose as measurements. Since these measurements have constant size (3 or 6 dimensions, respectively) our state estimator which is also responsible for the inter-sensor calibration has constant complexity.

The first part of the chapter, Section 3.1, is largely based on our work published in (Weiss and Siegwart (2011)). We focus on our core-sensor suite for MAV navigation: one IMU and one camera. Using this simple sensor setup we introduce our modular sensor fusion approach. This approach assumes the camera as a regular motion sensor – or, in other words, as a *black box* – directly yielding its scaled pose with respect to its own reference frame or yielding its scaled body-velocity. In this first part, we do not detail the issues of the visual position drifts yet. Rather,in Section 3.1.1, we design a simplified EKF framework for visual-inertial micro helicopter navigation tackling in particular the (inter-)sensor calibration as well as visual scale and attitude drifts. We discuss the detection of failure modes of the vision black box in Section 3.1.2.

In the second part, we use the core-sensor setup discussed in the first part of the chapter and extend it to a general multi-sensor framework. This part

is dedicated to the theoretical analysis of this framework. We do a non-linear observability analysis on the core system and identify its unobservable modes (Section 3.3). Based on this result, in Section 3.4, we discuss the addition of different sensors and their influence on the unobservable modes. We describe 4 base scenarios and add 3 sensors to the system. Each of the first three base scenarios is defined by a specific group of measurement types added to the core system. The last scenario is dedicated to the camera as a scaled body-velocity sensor as discussed in Section 2.2. It's observability analysis motivates the use of this map-less approach to bridge and initialize the map-based approach when the camera is used as a 6DoF pose sensor. The above provides the necessary understanding for Section 3.4.6 on how to add any *type* of sensor to the system. We summarize our findings on observable and unobservable modes using different sensor types with the propagation module of our core sensor suite (i.e. with the IMU) in a table. This table allows to concatenate different sensor types to reflect real world sensors.

We conclude the chapter in Section 3.5

3.1 Visual-Inertial MAV Navigation

A major issue of vision based MAV navigation using one camera as the only exteroceptive sensor is the recovery of the metric error between the desired vehicle pose and the actual pose. Only based on this, robust controllers can be designed. We discuss in this section the foundations of our metric state estimator based on the core-sensor suite.

Monocular approaches inherently suffer from the lack of metric scale. Apart from having a known visual pattern one can think of a variety of sensors to retrieve the absolute scale. While laser rangefinders are too heavy, infrared sensors are too sensitive to sun light and ultra sound sensors lack in their range in which they reliably can measure the distance. This reduces the choice to a second camera, a pressure sensor or an inertial sensor (i.e. IMU). However, the first loses its range measurement ability for scenes far away with respect to the stereo baseline. The second option is unreliable indoors (sudden pressure changes when closing doors, activating air conditioning etc). This leaves the IMU as logical choice. For fusing the IMU data with the visual input, a least square or an Extended Kalman Filter (EKF) approach are the choices at hand, as an Unscented Kalman Filter is too costly for on-board processing. One can imagine comparing the integrated inertial values to the derivated visual positions in a least squares setup (Kneip et al. (2011)). Despite the possible high speed of this approach, it lacks of robustness because of its sensitivity to outliers. Furthermore, the

least squares approach only estimates the visual scale, that is, we do not obtain an optimal estimation of the robot's velocity and position in order to control it efficiently. A more complete least squares approach was suggested in (Martinelli (2012); Martinelli et al. (2011)) where the authors include the velocity, attitude and bias in their calculations. The sensitivity to outliers and the lack of self-calibration remains. These drawbacks led us to an EKF method.

In contrast to previous EKF approaches, our method is independent of the underlying vision algorithm which estimates the camera poses. This has the advantage that the algorithm presented here operates at a constant computational complexity. This allows us to treat the visual framework as a black box and thus the approach is modular and widely applicable to existing (monocular) solutions. It can be used with any 6DoF pose estimation algorithm such as visual odometry or visual SLAM in monocular or stereo setups, ICP based range finder approaches (Kinect, 3D laser scanners, etc), or external estimators like Vicon, AR-Toolkit etc. In particular, we are independent of the computationally costly vision based EKF-SLAM for camera pose estimation. This makes our approach a very versatile solution to control a robot efficiently and at high frequency in metric position and velocity. Fig. 3.1 compares the standard approach with our method. Even though we treat the visual framework as a black box, we show how to detect failures and estimate drifts in it.

3.1.1 Extended Kalman Filter Framework

Considering the camera as an additional sensor to compensate for the temporal drift of the IMU, additional sensors like magnetometers or GPS can be seen as sensors to compensate for the spatial drift of the camera. Fig. 3.2 illustrates a non-exhaustive classification of different sensors. The robotics community showed that local navigation capabilities are sufficient for large scale robot applications (Furgale and Barfoot (2010)). However, in this work we aim at a general system that can easily be augmented with further sensors. We propose an approach to determine the capabilities and limitations of the applied sensor suite. We consider it thus as a possible and versatile standard for sensor fusion on power-on-and-go systems.

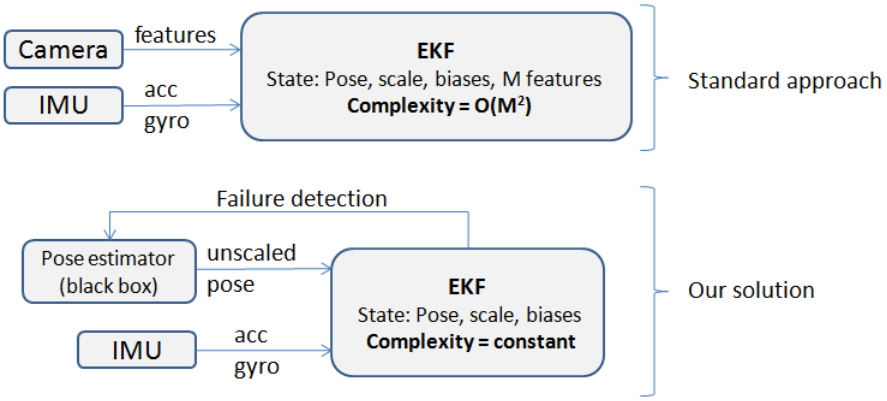


Figure 3.1: Standard tightly coupled solutions using an EKF SLAM approach (top schematic) have a computational complexity of at least $O(M^2)$ with M being the number of features. In contrast, our approach runs at constant computational complexity. Moreover, treating the pose estimation part as a black box, we are independent of the underlying pose estimation method. The whole framework has thus the complexity of the chosen pose estimator. As we show in this work, we can still detect failures and drifts of the black box. (Weiss and Siegwart (2011))

Core Setup and Sensor Model

In this paragraph we discuss our core-sensor setup which only consists of an IMU as the inertial propagation sensor and a camera as the visual update sensor. Further we discuss the sensor model of this setup. The inertial sensor yields the acceleration and rotational velocity in 3 axes. The visual sensor provides the system 3D position in undefined scale, and scale free attitude estimations with respect to its own visual frame. Fig. 3.3 shows the situation of the camera-IMU setup with its corresponding coordinate frames: In green, the transformation from the IMU frame into the camera frame contains the calibration parameters of the system. They are constant up to structural deformations. In red, the transformation from the world frame into the IMU frame is our robot pose of interest. In blue, the transformation from the visual frame into the camera is measured with unknown scale by the black boxed visual framework. In gray, the transformation between the visual frame and the world frame is constant for short periods of time. We introduce this transformation as rotational (q_v^w) and translational (p_v^w) drifts of the black boxed visual framework with respect to the world frame. The need for the introduction of these drift states and the world reference frame

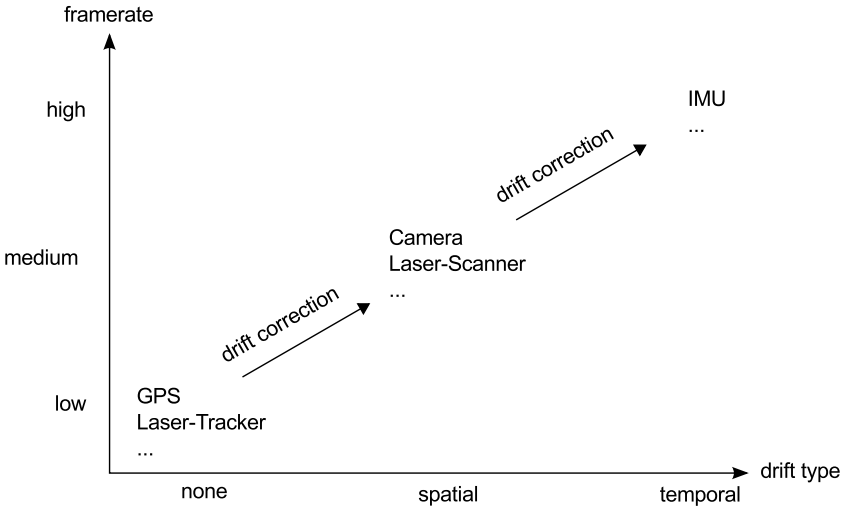


Figure 3.2: We illustrate a non-exhaustive classification of different sensors. A camera with visual or a laser-scanner with ICP based pose estimation could be considered as a sensor to eliminate the temporal drift of an IMU. Further, globally consistent sensors such as (laser) tracking systems or GPS could be considered to eliminate the spatial drift of the camera or laser-scanner. While not all robot applications need a globally consistent pose estimate, we consider it important to have one unifying approach for a self-calibrating pose estimator. The sensor suite is then the designer's choice, and the resulting capabilities and limitations are presented in this work.

respectively arises from two properties of our proposed system. First, the (possibly arbitrarily scaled and spatially drifting) pose input can be generated by any suitable algorithm or sensor. Our system treats this input as one coming from a black box. Second, the addition of multiple sensors requires an independent reference frame to which the sensors can be referred to.

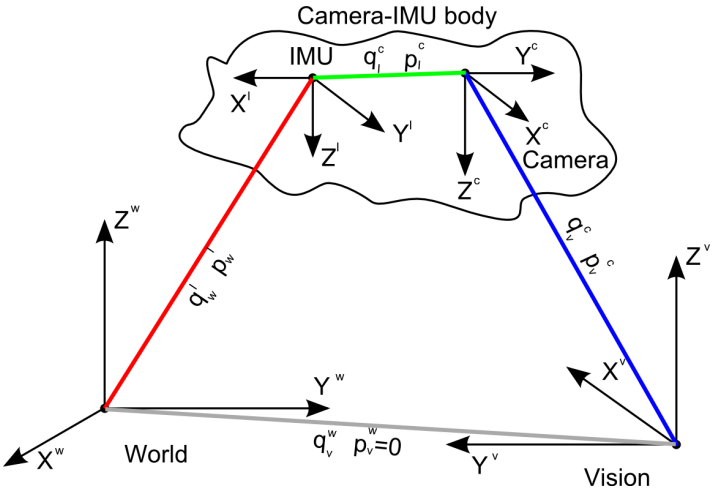


Figure 3.3: Visualization of the different coordinate frames in the setup. Between every frame there is a rotation q and translation p. Roughly constant values are in green, such as the IMU-cam transformation. Red values denote the variables used for robot control. Blue values are the measurements of the black boxed visual framework. In gray we denote the short term stationary transformations between the vision frame of reference and the world frame. Note that change in q_v^w reflects the angular drift whereas p_v^w reflects the translational drift in the visual framework with respect to the world frame. (Weiss and Siegwart (2011))

We assume that the inertial measurements contain a certain bias b and white Gaussian noise n. Thus, for the real angular velocities ω and the real accelerations a in the IMU frame we have

$$\omega = \omega_m - b_\omega - n_\omega \qquad a = a_m - b_a - n_a, \tag{3.1}$$

where the subscript m denotes the measured value. The dynamics of the non-static biases b are modeled as a random process

$$\dot{b_\omega} = n_{b_\omega} \qquad \dot{b_a} = n_{b_a}. \tag{3.2}$$

In summary, we have the following characteristics for the noise models in continuous time

$$E(n_a) = E(n_\omega) = E(n_{b_a}) = E(n_{b_w}) = 0 \tag{3.3}$$
$$E(n_\omega(t+\tau) * n_\omega^T(t)) = \sigma_\omega^2 \delta(\tau) \tag{3.4}$$
$$E(n_{b_\omega}(t+\tau) * n_{b_\omega}^T(t)) = \sigma_{b_\omega}^2 \delta(\tau) \tag{3.5}$$
$$E(n_a(t+\tau) * n_a^T(t)) = \sigma_a^2 \delta(\tau) \tag{3.6}$$
$$E(n_{b_a}(t+\tau) * n_{b_a}^T(t)) = \sigma_{b_a}^2 \delta(\tau) \tag{3.7}$$

Knowing that $[\delta(x)] = \frac{1}{[x]}$ (where $[x]$ denotes the SI units of x) and that the units of the white Gaussian noise are equal to the measured sensor values, in the continuous time case we have the units

$$\left[\sigma_\omega^2\right] = \frac{\left[\sigma_\omega^2 \delta(\tau)\right]}{[\delta(\tau)]} = \frac{rad^2}{sec} \tag{3.8}$$
$$\left[\sigma_{b_\omega}^2\right] = \frac{\left[\sigma_{b_\omega}^2 \delta(\tau)\right]}{[\delta(\tau)]} = \frac{rad^2}{sec^3} \tag{3.9}$$
$$\left[\sigma_a^2\right] = \frac{\left[\sigma_a^2 \delta(\tau)\right]}{[\delta(\tau)]} = \frac{m^2}{sec^3} \tag{3.10}$$
$$\left[\sigma_{b_a}^2\right] = \frac{\left[\sigma_{b_a}^2 \delta(\tau)\right]}{[\delta(\tau)]} = \frac{m^2}{sec^5} \tag{3.11}$$

These values are generally given by the manufacturer. For discrete time steps, as it will be applied in the filter, we need to convert the values according to their units:

$${}_d\sigma_\omega^2 = \frac{\sigma_\omega^2}{\Delta t} \qquad {}_d\sigma_{b_\omega}^2 = \sigma_{b_\omega}^2 * \Delta t \tag{3.12}$$
$${}_d\sigma_a^2 = \frac{\sigma_a^2}{\Delta t} \qquad {}_d\sigma_{b_a}^2 = \sigma_{b_a}^2 * \Delta t \tag{3.13}$$

Using the manufacturer's data and the transformation equation of equations (3.12) to (3.13) we obtain a well dimensioned model for the process noise in the filter.

State Representation

The state of the filter is composed of the position p_w^i of the IMU in the world frame W, its velocity v_w^i, and its attitude quaternion q_w^i describing a

rotation from the world frame W into the IMU frame I. We also add the gyro and acceleration biases b_ω and b_a as well as the visual scale factor λ. The calibration states are the rotation from the IMU frame into the camera frame q_i^c, and the position of the camera center in the IMU frame p_i^c. Additionally, we include the drifts between the black boxed visual frame V and the fixed world frame W. The rotational drifts are reflected in q_v^w and the translational ones in p_v^w. We do not include the gravity vector in the state space, since our world reference frame is aligned with it and thus known. The gravity vector has been estimated in other work (such as in (Kelly and Sukhatme (2011))) because the reference frame was the drifting and unaligned vision frame. Our entire state yields a 31-elements state vector X:

$$X = \{p_w^{i^T} \; v_w^{i^T} \; q_w^{i^T} \; b_\omega^T \; b_a^T \; \lambda \; p_i^c \; q_i^c \; p_v^w \; q_v^w\} \tag{3.14}$$

The following differential equations govern the state:

$$\begin{aligned} \dot{p}_w^i &= v_w^i && (3.15)\\ \dot{v}_w^i &= C_{(q_w^i)}^T(a_m - b_a - n_a) - g && (3.16)\\ \dot{q}_w^i &= \frac{1}{2}\Omega(\omega_m - b_\omega - n_\omega)q_w^i && (3.17) \end{aligned}$$

$$\dot{b}_\omega = n_{b_\omega} \quad \dot{b}_a = n_{b_a} \quad \dot{\lambda} = 0 \tag{3.18}$$

$$\dot{p_i^c} = 0 \quad \dot{q_i^c} = 0 \quad \dot{p_v^w} = 0 \quad \dot{q_v^w} = 0 \tag{3.19}$$

$C_{(q)}$ is the rotational matrix corresponding to the quaternion q, g is the gravity vector in the world frame, and $\Omega(\omega)$ is the quaternion-multiplication matrix of ω. We assume the scale and visual frame V to drift spatially and not temporally. Thus we apply a zero motion model. Such states need special attention in the implementation of the system because their certainty has to be actively adapted upon the specific spatial event. For example, when using a key-frame based visual framework to calculate the camera pose, the covariance of the visual scale λ and the visual frame drift states p_v^w, q_v^w should be augmented upon key-frame creation only and according to the visual algorithm's performance. Taking the expectations of the above derivatives and the beforehand discussed noise model for the filter state propagation, we obtain

$$\begin{aligned}
\dot{\hat{p}}_w^i &= \hat{v}_w^i && (3.20)\\
\dot{\hat{v}}_w^i &= C_{(\hat{q}_w^i)}^T (a_m - \hat{b}_a) - g && (3.21)\\
\dot{\hat{q}}_w^i &= \frac{1}{2}\Omega(\omega_m - \hat{b}_\omega)\hat{q}_w^i && (3.22)
\end{aligned}$$

$$\dot{\hat{b}}_\omega = 0 \quad \dot{\hat{b}}_a = 0 \quad \dot{\hat{\lambda}} = 0 \qquad (3.23)$$

$$\dot{\hat{p}}_i^c = 0 \quad \dot{\hat{q}}_i^c = 0 \quad \dot{\hat{p}}_v^w = 0 \quad \dot{\hat{q}}_v^w = 0. \qquad (3.24)$$

Error State Representation

In the above described state representation, we use quaternions as attitude description. It is common that in such a case we represent the error and its covariance not in terms of an arithmetic difference but with the aid of an error quaternion. This increases numerical stability and handles the quaternion in its minimal representation (Trawny and Roumeliotis (2005)). Therefor, we define the 28-elements error state vector

$$\tilde{x} = \{\Delta p_w^{i^T} \ \Delta v_w^{i^T} \ \delta\theta_w^{i^T} \ \Delta b_\omega^T \ \Delta b_a^T \ \Delta\lambda \ \Delta p_i^{c^T} \ \delta\theta_i^{c^T} \ \Delta p_v^{w^T} \ \delta\theta_v^{w^T}\} \qquad (3.25)$$

as the difference of an estimate $\hat{x}$ to its quantity x, i.e. $\tilde{x} = x - \hat{x}$. We apply this to all state variables except the error quaternions which are defined as follows

$$\begin{aligned}
\delta q_w^i &= q_w^i \otimes \hat{q}_w^{i^{-1}} \approx [\frac{1}{2}\delta\theta_w^{i^T} \ 1]^T && (3.26)\\
\delta q_i^c &= q_i^c \otimes \hat{q}_i^{c^{-1}} \approx [\frac{1}{2}\delta\theta_i^{c^T} \ 1]^T && (3.27)\\
\delta q_v^w &= q_v^w \otimes \hat{q}_v^{w^{-1}} \approx [\frac{1}{2}\delta\theta_v^{w^T} \ 1]^T && (3.28)
\end{aligned}$$

The differential equations for the continuous time error state are

$$\begin{aligned}
\Delta\dot{p}_w^i &= \Delta v_w^i && (3.29)\\
\Delta\dot{v}_w^i &= -C_{(\hat{q}_w^i)}^T \lfloor \hat{a}_\times \rfloor \delta\theta - C_{(\hat{q}_w^i)}^T \Delta b_a - C_{(\hat{q}_w^i)}^T n_a && (3.30)\\
\dot{\delta\theta}_w^i &= -\lfloor \hat{\omega}_\times \rfloor \delta\theta - \Delta b_\omega - n_\omega && (3.31)
\end{aligned}$$

$$\Delta\dot{b}_\omega = n_{b_\omega} \quad \Delta\dot{b}_a = n_{b_a} \quad \Delta\dot{\lambda} = 0 \tag{3.32}$$

$$\Delta\dot{p}_i^c = 0 \quad \delta\dot{q}_i^c = 0 \quad \Delta\dot{p}_v^w = 0 \quad \delta\dot{q}_v^w = 0 \tag{3.33}$$

with $\hat{\omega} = \omega_m - \hat{b_\omega}$ and $\hat{a} = a_m - \hat{b_a}$. This can be summarized to the linearized continuous-time error state equation

$$\dot{\tilde{x}} = F_c\tilde{x} + G_c n \tag{3.34}$$

with n being the noise vector $n = [n_a^T, n_{b_a}^T, n_\omega^T, n_{b_\omega}^T]^T$. In the solution presented here, we are in particular interested in the speed of the algorithm. This is why we assume F_c and G_c to be constant over the integration time step between two consecutive state propagations. For the discretization we may therefor write

$$F_d = \exp(F_c\Delta t) = \mathbf{I_d} + F_c\Delta t + \frac{1}{2!}F_c^2\Delta t^2 + \dots \tag{3.35}$$

Careful analysis of the matrix exponents reveals a repetitive and sparse structure. This allows us to express F_d exactly without approximation and to use sparse matrix operations later in the implementation of the filter. The matrix F_d has the following structure:

$$F_d = \begin{bmatrix} \mathbf{I_{d_3}} & \Delta t & A & B & -C^T_{(\hat{q}^i_w)}\frac{\Delta t^2}{2} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{I_{d_3}} & C & D & -C^T_{(\hat{q}^i_w)}\Delta t & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & E & F & \mathbf{0_3} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_3} & \mathbf{0_{3x13}} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_{3x13}} \\ \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{I_{d_{13}}} \end{bmatrix}$$

We now use the small angle approximation for which $|\omega| \to 0$, apply de l'Hopital rule and obtain for the six matrix blocks A, B, C, D, E, F a compact solution (Trawny and Roumeliotis (2005)):

$$
\begin{aligned}
A &= -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\frac{\Delta t^2}{2} - \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor + \frac{\Delta t^4}{4!} \lfloor \omega_\times \rfloor^2 \right) \\
B &= -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\frac{-\Delta t^3}{3!} + \frac{\Delta t^4}{4!} \lfloor \omega_\times \rfloor - \frac{\Delta t^5}{5!} \lfloor \omega_\times \rfloor^2 \right) \\
C &= -C^T_{(\hat{q}^i_w)} \lfloor \hat{a}_\times \rfloor \left(\Delta t - \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor + \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor^2 \right) \\
D &= -A \\
E &= \mathbf{I_d} - \Delta t \lfloor \omega_\times \rfloor + \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor^2 \\
F &= -\Delta t + \frac{\Delta t^2}{2!} \lfloor \omega_\times \rfloor - \frac{\Delta t^3}{3!} \lfloor \omega_\times \rfloor^2
\end{aligned}
$$

G_c is then from (3.34)

$$
G_c = \begin{bmatrix}
\mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\
-C^T_{(\hat{q}^i_w)} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\
\mathbf{0_3} & \mathbf{0_3} & -\mathbf{I_{d_3}} & \mathbf{0_3} \\
\mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_{d_3}} \\
\mathbf{0_3} & \mathbf{I_{d_3}} & \mathbf{0_3} & \mathbf{0_3} \\
\mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}} & \mathbf{0_{13x3}}
\end{bmatrix}
$$

We can now derive the discrete time covariance matrix Q_d (Maybeck (1979)) as

$$
Q_d = \int_{\Delta t} F_d(\tau) G_c Q_c G_c^T F_d(\tau)^T d\tau, \tag{3.36}
$$

with Q_c being the continuous time system noise covariance matrix $Q_c = \text{diag}(\sigma^2_{n_a}, \sigma^2_{n_{b_a}}, \sigma^2_{n_\omega}, \sigma^2_{n_{b_\omega}})$.

With the discretized error state propagation and error process noise covariance matrices we can propagate the state as follows:

1. propagate the state variables according to their differential equations (3.20) to (3.24). For the quaternion, we used the first order integration described in (Trawny and Roumeliotis (2005)).

2. calculate F_d and Q_d

3. compute the propagated state covariance matrix according to the filter equation

$$
P_{k+1|k} = F_d P_{k|k} F_d^T + Q_d \tag{3.37}
$$

Measurement

While body accelerations and angular velocities are used for the state propagation, the mentioned (camera) pose measurement is used for the filter update. That is, we account for the temporal drift of the sensor used in the propagation phase. The pose measurement may drift spatially in its attitude and position. For easier understanding, we neglect the position drift for the moment and assume the measurement only drifts in all three angular directions. Mathematically, this means that p_v^w is zero and can be neglected in the filter equations. Intuitively, this means that our world frame is no longer fix, but drifts in position with the drift of the pose measurement. In the remainder of the section, we keep this position drifts in the discussion but only consider them in Section 3.4.3.

Measurement: Covariances

For the Filter update, we first estimate the covariance of the measurement noise. For example, the covariance of the camera pose estimation in an EKF-SLAM is straight forward. For calculating the covariance of a pose estimation using the five-point algorithm or similar we refer to (Beder and Steffen (2006)). Pose estimations from nonlinear optimization steps in key-frame based solutions are more complex: We suggest to use the estimation from (Beder and Steffen (2006)) for the first two key-frames. For the following ones, we suggest the method of (Eudes and Lhuillier (2009)) defining the N closest key-frames as loose and the rest (minimum the first two) as fixed. As an approximation, we assume the actual pose to have the same uncertainty as the distance weighted mean of the uncertainties of the nearest N lose key-frames. Note that N is a design parameter to ensure fast processing time. The estimation of the pose covariance is not as trivial if the used pose estimation algorithm is completely closed source and only providing the pose without any accuracy information. However, common existing pose estimation algorithms such as (Klein and Murray (2007)) provide some sort of accuracy information which can then be used for approximations.

We distinguish between the position n_p and rotational n_q measurement noise. This yields the six-vector

$$n_m = [n_p \; n_q]^T \tag{3.38}$$

with its measurement-covariance matrix R. Note that this noise model is an approximation which is needed in the Kalman Filter theory. In reality, neither the noise in position nor in rotation is temporally uncorrelated or has a Gaussian nature. Such approximations are one reason why the implemented

linearized and discretized system has to be tested under real conditions to ensure its functioning (see Chapter 4).

Measurement: Model

For the camera position measurement p_v^c we obtain from the visual algorithm, we have the following measurement model

$$z_p = p_v^c = C_{(q_v^w)}^T (p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + n_p \tag{3.39}$$

with $C_{(q_w^i)}$ as the IMU's attitude and $C_{(q_v^w)}$ the rotation from the visual frame into the world frame.

We define the position error as

$$\begin{array}{rcl} \tilde{z}_p & = & z_p - \hat{z}_p \\ & = & C_{(q_v^w)}^T (p_w^i + C_{(q_w^i)}^T p_i^c)\lambda + n_p - \\ & & C_{(\hat{q}_v^w)}^T (\hat{p}_w^i + C_{(\hat{q}_w^i)}^T \hat{p}_i^c)\hat{\lambda} \end{array} \tag{3.40}$$

which can be linearized to

$$\tilde{z}_{p_l} = H_p \tilde{x}, \tag{3.41}$$

with

$$H_p = \begin{bmatrix} C_{(\hat{q}_v^w)}^T \hat{\lambda} \\ \mathbf{0_3} \\ -C_{(\hat{q}_v^w)}^T C_{(\hat{q}_w^i)}^T \left\lfloor \hat{p_i^c} \right\rfloor \hat{\lambda} \\ \mathbf{0_3} \\ \mathbf{0_3} \\ C_{(\hat{q}_v^w)}^T C_{(\hat{q}_w^i)}^T \hat{p_i^c} + C_{(\hat{q}_v^w)}^T \hat{p} \\ C_{(\hat{q}_v^w)}^T C_{(\hat{q}_w^i)}^T \hat{\lambda} \\ \mathbf{0_3} \\ -C_{(\hat{q}_v^w)}^T \left\lfloor (p_w^i + C_{(\hat{q}_w^i)}^T p_i^c)\lambda) \right\rfloor \end{bmatrix}^T$$

using the definition of the error quaternion

$$q_w^i = \delta q_w^i \otimes \hat{q}_w^i \tag{3.42}$$

$$C_{(q_w^i)} = C_{(q_i^i)} C_{(q_w^i)} \tag{3.43}$$

$$C_{(q_i^i)} \approx \mathbf{I_d} - \left\lfloor \delta\theta_{\mathbf{w}_\times}^{\mathbf{i}} \right\rfloor. \tag{3.44}$$

For the rotation measurement, we again apply the notion of an error quaternion. The vision algorithm yields the rotation from the vision frame into the camera frame q_v^c. We can model this as

$$z_q = q_v^c = q_i^c \otimes q_w^i \otimes q_v^w, \tag{3.45}$$

which yields the error measurement

$$\begin{aligned} \tilde{z}_q &= z_q - \hat{z}_q \\ &= q_i^c \otimes q_w^i \otimes q_v^w \otimes (q_i^c \otimes \hat{q}_w^i \otimes \hat{q}_v^w)^{-1} \\ &= H_q^{wi} \delta q_w^i = H_q^{ic} \delta q_i^c = H_q^{vw} \delta q_v^w, \end{aligned} \tag{3.46}$$

where H_q^{wi}, H_q^{ic} and H_q^{vw} are the matrices when the error measurement is linearized versus the filter error states δq_w^i, δq_i^c, and δq_v^w, respectively. Finally, the measurements can be stacked together as

$$\begin{aligned} \begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} &= \begin{bmatrix} & H_p & \\ \mathbf{0_{3x6}} & \tilde{H}_q^{wi} & \mathbf{0_{3x10}} \; \tilde{H}_q^{ic} \; \tilde{H}_q^{vw} \end{bmatrix} \tilde{x} \\ \tilde{\mathbf{z}} &= \mathbf{H}\tilde{\mathbf{x}} \end{aligned} \tag{3.47}$$

using the approximation of $H_q^{xy} = \begin{pmatrix} 1 & 0_{1x3} \\ 0_{3x1} & \tilde{H}_q^{xy} \end{pmatrix}$. This is justified since the expectation of the error quaternions δq_w^i and δq_i^c are unit quaternions.

Measurement: Update

Once we obtained the measurement matrix $\mathbf{H}$ we can update our estimate according to the well known Kalman Filter procedure:

1. compute the residual $\tilde{z} = z - \hat{z}$
2. compute the innovation $S = HPH^T + R$
3. compute the Kalman gain $K = PH^T S^{-1}$
4. compute the correction $\hat{\tilde{x}} = K\tilde{z}$

From the correction $\hat{\tilde{x}}$ we can compute the updated state variables in our state X. The error quaternion is calculated by following (3.26) and ensuring its unit length. The error state covariance is updated as following

$$P_{k+1|k+1} = (\mathbf{I_d} - KH)P_{k+1|k}(\mathbf{I_d} - KH)^T + KRK^T \tag{3.48}$$

This concludes the basic filter design and its implementation in an EKF framework. As we will see in the next section, this design contains as such unobservable states. We discuss this issue in detail and find the interconnectivity between the unobservable states in order to find adequate solutions.

3.1.2 False Pose Estimate Detection

Unlike tightly coupled approaches, with our solution, we can treat the visual part (i.e. the visual pose estimation) as a black box. Of course, one desires to detect failures and drifts of the black box to ensure ongoing functionality of the whole framework. The scale drift is handled by the scale estimate in real time. We explain the detection of failures of the vision part in the following.

We note that in Section 3.1.1 we considered the rotation between the inertial world frame and the frame of the visual framework q_v^w as constant during a filter step. At each filter step k we can measure this rotation as

$$q_v^w(k) = \hat{q}_w^{i^{-1}}(k) \otimes \hat{q}_i^{c^{-1}}(k) \otimes q_v^c(k) \tag{3.49}$$

Since the drift is slow compared to the filter and measurement frequency, we can smooth a sequence of measurements of $q_v^w(k)$. We suggest a median filter as the vision part is better modeled with non-zero mean outlier jumps. The estimation of the rotation between inertial and visual frame $\hat{q}_v^w(k)$ using a window of size N is then

$$\hat{q}_v^w(k) = \text{med}(q_v^w(i)) \quad i = k - N \to k \tag{3.50}$$

Due to the fact that the drift is slow, we can identify abrupt jumps in the measured orientation $q_v^w(k)$ with respect to the smoothed estimate $\hat{q}_v^w(k)$ as failures of the visual pose estimation (see also Section 2.1.2). A typical plot of the measurement of $q_v^w(k)$ is shown in Fig. 3.4. The sequences where the visual part failed to estimate a correct pose are clearly visible. As soon as a measurement $q_v^w(k)$ lies outside the 3σ error bounds of the past M estimates $\hat{q}_v^w(k)$ it is considered as false pose estimate. Note that, analyzing the rotation for wrong measurements, this is less sensitive to loop closure jumps. Usually these jumps affect greatly the position but less the attitude.

During such failure periods neither $\hat{q}_v^w(k)$ nor the biases and scale are updated since the measured data is corrupted. Note that we do update the IMU orientation, the velocity and position to bound the error during long dropouts. Low cost accelerometers suffer from large drifts and should not be integrated over longer periods of time without updates. In contrast, gyroscope integration is fairly robust also over longer periods of time. Thus

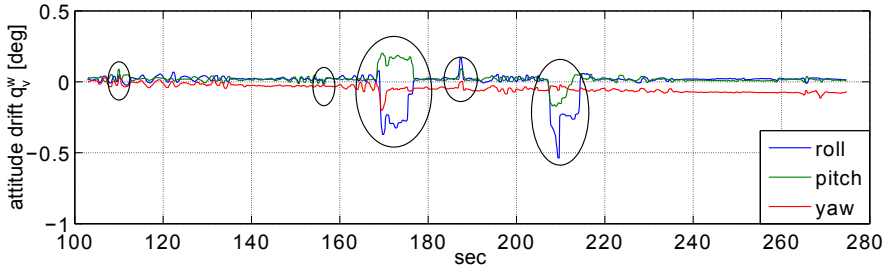


Figure 3.4: Roll, pitch, yaw representation of the calculated rotation between the black-box visual frame and the world frame. Note the five encircled areas which depict a failure of the vision algorithm. Even though we treat the vision algorithm as a black box we can clearly identify failures. Note the apparent drift in yaw, we discuss this issue in Section 3.4. (Weiss and Siegwart (2011))

we increase the visual measurement noise for the position much less than for the attitude. The increasing of the measurement noise during failure sequences bounds the filter error yet trusts the simple integration to a large extent. As figure 3.5 shows, the attitude is still estimated reliably.

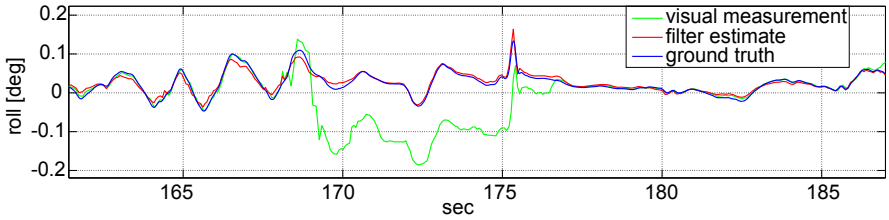


Figure 3.5: Example for the roll angle estimate evolution during a failure of the black-box visual framework. During a failure mode biases and scale are not updated. The measurement noise for the attitude is highly increased as gyroscopes are reliable for long term integration. The noise for the position measurement is less increased, because accelerometers suffer from larger drifts. This results in an slight erroneous position estimate of the filter but still very reliable attitude estimate of the MAV. (Weiss and Siegwart (2011))

With the above approaches we showed both, the handling of scale drift and failures of the visual part treated as a black box.

3.2 Modular Multisensor Systems

In the previous section of this chapter we discussed our core-sensor suite consisting of a camera and an IMU for vision based navigation for micro helicopters. In essence this approach is sufficient for locally stable navigation as we detail in (Achtelik et al. (2011b); Weiss and Siegwart (2011)). Since the IMU senses gravity but the camera pose suffers from 6DoF spatial drift, one might assume that our core-sensor suite only recovers the MAV state up to a global yaw and global 3D position.

In other words, neglecting the visual position and yaw drift, we already have a full system running in real-time and on-board. We not nonly estimate the MAVs state but also calibrate the inter-sensor states and estimate the visual roll, pitch, and scale drift, and the IMU biases. The sensor suite of only one IMU and one camera weights 20 g and turns the MAV into a real-world locally stable power-on-and-go system (Weiss and Siegwart (2011)).

For many tasks, neglecting position and yaw drift can be legitimate if an appropriate path planner is applied (e.g. visual homing). Nevertheless, we aim at a modular and versatile system that can be expanded with a multitude of different sensors while still avoiding tedious calibration procedures. In the following, we use the findings on our core-sensor suite in the previous part and extend it to a real-time, on-board, and self-calibrating multi-sensor system to account for the different vision drifts.

Multi-sensor systems for long-term navigation have three main requirements to the pose estimation algorithm. First, it should be scalable to the environment in question. Second, it should be extendable with additional (drift free or slowly drifting) sensors. Third, the algorithm should account for any misalignment of the sensors occurring during the navigation.

All three requirements are met by considering each sensor as a separate unit to estimate the robot's pose or part of it, and fusing each output in a statistically optimal way. Such type of system is called loosely-coupled. The discussed decoupling of the vision part allows us to employ any existing and computationally cheap visual pose estimator in a black-box style. This tackles the first requirement. The actual robot position is defined by the IMU center expressed in the world frame, and the drift of the camera's pose estimate is modeled as a rotational and translational offset between the vision and the world reference frame.

Further sensors can be added and referred to the world frame. A sensor yielding a fix attitude such as a magnetometer will compensate for the angular yaw drift of the entire vision-inertial system, and align our world frame to a constant north. Likewise, a non-drifting position sensor such as a GPS-

receiver will account for angular and especially translational drift of the vision reference frame and thus yield a drift free robot pose estimate. This tackles the second requirement.

We use differential geometry to analyze the system's observability and detect the observable and unobservable variables. Based on these results, we discuss different scenarios for vision-inertial navigation with additional sensors. Despite of the fact that some drifts might not be accounted for depending on which sensors are used, we will see in Section 3.4 that in each setup all inter-sensor transformations are observable. To conclude, the system is continuously self-calibrating all inter-sensor transformations between the mounted vision, inertial and other additional sensors. This tackles the third requirement.

As we discuss in this work, drifts in the estimated pose are determined within the sensor suite's limits. To the best of our knowledge, this approach is more general than previous work in literature and represents a major step toward power-on-and-go robot systems. Since this approach is modular and different sensors can be added or neglected, we suggest this approach as standard approach for self-calibrating vision-inertial systems in robot navigation.

At this point we note that for such filter system, we consider three tracks of analysis. We name the first as the theoretical analysis of the theoretical system. That is, we analyze the non-linear, time continuous system. We fully explore this track in this part of this chapter. We consider the second track as the theoretical analysis of the practical system. This involves the analysis of the linearized, time discrete system. There, questions arise about the maximal allowed noise level or minimally needed frequency of a sensor to keep the system stable. Also of interest in this track is the question of how well the states are estimated, or, equivalently how well are they converged yet. While these questions certainly are of interest, we do not consider them in detail, but point to existing work (Davison (1998); Mourikis (June 2008)). The third track is the practical test of the implemented system. These tests are performed in Chapter 4.

3.3 Observability Analysis and Discussion

In this section we do the observability analysis for our core-sensor suite. We augment the level of abstraction by considering the *type* of information a sensor yields, rather than considering a specific sensor itself. We use one sensor for body acceleration and angular velocity readings (i.e. IMU). We use this information to propagate the filter state. A second sensor yielding a spatially drifting attitude and position with respect to its own reference

frame is used to correct for the temporal drift of the propagated state. In our case, this is a camera. In the following, we analyze the observability of this core system using tools of differential geometry. We apply the non-linear observability analysis proposed by Herman and Krener in (Hermann and Krener (1977)). We refer to the work of Mirzaei (Mirzaei and Roumeliotis (2007)) and Kelly (Kelly and Sukhatme (2011)) and Martinelli (Martinelli (2011)) for details about how to apply this method to a system similar to ours. We do not recommend an observability analysis in the spirit of (Jones (2009)) since we aim at an easily extensible method for additional sensors. Moreover, the authors of (Jones (2009)) did not consider the bias terms in their observability analysis in order to reduce the complexity.

For the properties and notation on quaternions we refer to (Trawny and Roumeliotis (2005)). For the notation of *Lie derivatives* we refer to (Kelly and Sukhatme (2011)). The authors give a short but very comprehensive overview about this mathematical tool for non-linear observability analysis. In short, the Lie derivative of h with respect to f at $x \in M$ is

$$L_f h(x) = \nabla_f h(x) = \frac{\partial h(x)}{\partial x} f(x). \tag{3.51}$$

The recursive definition is then

$$L_g L_f h(x) = \frac{\partial L_f h(x)}{\partial x} g(x). \tag{3.52}$$

If we take k-th derivative of h along f, we write

$$L_f^k h(x) \tag{3.53}$$

and apply the recursive definition. Note that, here, $\frac{\partial h(x)}{\partial x}$ is a row vector whereas f is a column vector. $L_f h(x)$ is thus a scalar.

3.3.1 Non-Linear Observability Analysis

Our goal is to show *locally weakly observability* of the proposed system according to (Hermann and Krener (1977)). For this analysis, we work on the state variables and not on the error state. The definition of the error state is an approximation where second and higher order terms are discarded under the assumption of a small error state. [1] In contrast, the observability

[1] The small angle approximation is used in the derivations of Jacobians, where it allows to 'ignore' second (and higher) order terms. Similarly for additive error, second and higher order terms are discarded under the assumption that error-state is small. This results in a (linear) approximation to the nonlinear system, and as such, we are discarding some information that may be useful to prove the system's observability.

analysis on the full nonlinear system prevents information loss.

We first define the core system in its control affine form. We still assume $p_v^w = 0$ and discuss this later in this section.

$$\begin{bmatrix} \dot{p}_w^i \\ \dot{v}_w^i \\ \dot{q}_w^i \\ \dot{b}_\omega \\ \dot{b}_a \\ \dot{\lambda} \\ \dot{p}_i^c \\ \dot{q}_i^c \\ \dot{q}_v^w \end{bmatrix} = \underbrace{\begin{bmatrix} v_w^i \\ -C^T_{(q_w^i)} b_a - g \\ 0.5 \Xi_{(q_w^i)} b_\omega \\ 0_{3\times 1} \\ 0_{3\times 1} \\ 0 \\ 0_{3\times 1} \\ 0_{4\times 1} \\ 0_{4\times 1} \end{bmatrix}}_{f_0} + \underbrace{\begin{bmatrix} 0_{3\times 3} \\ 0_{3\times 3} \\ 0.5 \Xi_{(q_w^i)} \\ 0_{3\times 3} \\ 0_{3\times 3} \\ 0_{1\times 3} \\ 0_{3\times 3} \\ 0_{4\times 3} \\ 0_{4\times 3} \end{bmatrix}}_{f_1} \omega_m + \underbrace{\begin{bmatrix} 0_{3\times 3} \\ C^T_{(q_w^i)} \\ 0_{4\times 3} \\ 0_{3\times 3} \\ 0_{3\times 3} \\ 0_{1\times 3} \\ 0_{3\times 3} \\ 0_{4\times 3} \\ 0_{4\times 3} \end{bmatrix}}_{f_2} a_m \tag{3.54}$$

where $\Xi_{(q)}$ is the quaternion multiplication matrix for the quaternion of rotation q and we have $\dot{q} = \Omega(\omega) q = \Xi_{(q)} \omega$.

Note that (3.54) is very similar to the one stated in (Kelly and Sukhatme (2011)). The additional states q_v^w, however, will change the measurement equations and the observability analysis fundamentally. The measurements can be summarized as

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{bmatrix} = \begin{bmatrix} C^T_{(q_v^w)} (p_w^i + C^T_{(q_w^i)} p_i^c) \lambda \\ q_i^c \otimes q_w^i \otimes q_v^w \\ {q_w^i}^T q_w^i \\ {q_i^c}^T q_i^c \\ {q_v^w}^T q_v^w \end{bmatrix}, \tag{3.55}$$

where h_1 expresses the visual position measurement p_v^c, h_2 expresses the visual attitude measurement q_v^c, and h_3 to h_5 are the unit norm constraints for each rotation quaternion in the system.

We denote the gradient with respect to the state variables of the zero order Lie derivative of h_n as $\nabla L^0 h_n$, and we denote the gradient of its first order derivative with respect to f_m as $\nabla L^1_{f_m} h_n$. Following the suggestions of Kelly in (Kelly and Sukhatme (2011)), we obtain the observability matrix $\mathcal{O}$

$$
\begin{bmatrix}
\nabla L^0 h_1 \\ \nabla L^0 h_2 \\ \nabla L^0 h_3 \\ \nabla L^0 h_4 \\ \nabla L^0 h_5 \\
\nabla L^1_{f_0} h_1 \\ \nabla L^1_{f_0} h_2 \\ \nabla L^1_{f_1} h_1 \\ \nabla L^1_{f_1} h_2 \\
\nabla L^2_{f_0 f_0} h_1 \\ \nabla L_{f_0 f_2} h_1 \\ \nabla L_{f_0 f_0 f_2} h_1
\end{bmatrix}
=
\begin{bmatrix}
\overbrace{}^{p_w^i} & \overbrace{}^{v_w^i} & \overbrace{}^{q_w^i} & \overbrace{}^{b_\omega} & \overbrace{}^{b_a} \\
C_v^{w\mathrm{T}}\lambda & 0 & [a_{1,3}] & 0 & 0 \\
0 & 0 & [a_{2,3}] & 0 & 0 \\
0 & 0 & 2q_w^{i\mathrm{T}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & C_v^{w\mathrm{T}}\lambda & [a_{6,3}] & [a_{6,4}] & 0 \\
0 & 0 & [a_{7,3}] & [a_{7,4}] & 0 \\
0 & 0 & [a_{8,3}] & 0 & 0 \\
0 & 0 & [a_{9,3}] & 0 & 0 \\
0 & 0 & [a_{10,3}] & [a_{10,4}] & [a_{10,5}] \\
0 & 0 & [a_{11,3}] & 0 & 0 \\
0 & 0 & [a_{12,3}] & [a_{12,4}] & [a_{12,5}]
\end{bmatrix}
$$

$$
\begin{bmatrix}
[a_{1,6}] & [a_{1,7}] & 0 & [a_{1,9}] \\
0 & 0 & [a_{2,8}] & [a_{2,9}] \\
0 & 0 & 0 & 0 \\
0 & 0 & 2q_i^{c\mathrm{T}} & 0 \\
0 & 0 & 0 & 2q_v^{w\mathrm{T}} \\
\begin{bmatrix} a_{6,6} \end{bmatrix} & [a_{6,7}] & 0 & [a_{6,9}] \\
0 & 0 & [a_{7,8}] & [a_{7,9}] \\
\begin{bmatrix} a_{8,6} \end{bmatrix} & [a_{8,7}] & 0 & [a_{8,9}] \\
0 & 0 & [a_{9,8}] & [a_{9,9}] \\
\begin{bmatrix} a_{10,6} \end{bmatrix} & [a_{10,7}] & 0 & [a_{10,9}] \\
\begin{bmatrix} a_{11,6} \end{bmatrix} & 0 & 0 & [a_{11,9}] \\
\begin{bmatrix} a_{12,6} \end{bmatrix} & [a_{12,7}] & 0 & [a_{12,9}]
\end{bmatrix}
\begin{matrix}
\underbrace{}_{\lambda} & \underbrace{}_{p_i^c} & \underbrace{}_{q_i^c} & \underbrace{}_{q_v^w}
\end{matrix}
$$

using the variables $a_{i,j}$ for better legibility.

Because of the omnipresent interaction of q_v^w in the matrix elements, a stepwise rank test as in (Kelly and Sukhatme (2011)) is not feasible. Moreover, symbolic rank tests on large matrices are known to be an issue. Even tools like Mathematica, Maple or Matlab fail on these tasks. We can see, however, that $\nabla L^0 h_1$ and $\nabla L^1_{f_0} h_1$ are responsible for the observability of the states p_w^i and v_w^i respectively. Only they have entries in the specific locations. Since the elements $a_{1,1}$ and $a_{6,2}$ are 3×3 linear independent matrices, each add 3 ranks to the matrix. Thus we analyze the observability of the remaining states with the remaining rows and columns only. Using Mathematica, we find that the sub-matrix $\bar{\mathcal{O}}$ consisting of rows 2-5,7-11 and columns 3-9 of $\mathcal{O}$ is rank deficient. This means that our system as it is now is not observable. In fact, the system has rank 27 instead of column full rank 28 (apart of ignoring the three position drift states p_v^w). Intuitively, the one unobservable dimension represents the absolute heading of the system. Since there is no reference to global yaw, this state is unobservable and we can be sure that even higher order Lie derivatives will not render $\mathcal{O}$ full column rank. The discussion of the unobservable states will be done in the following.

3.3.2 Discussion of the Unobservable States

The knowledge that we are missing only one single rank motivates to analyze the issue in more detail. Martinelli proposed in (Martinelli (2011)) a method to quantitatively describe the inter-dependencies of unobservable states. By analyzing the system on its continuous symmetries and corresponding indistinguishable regions, the method can be used to define an observable joint state containing the unobservable single states. To this end, we calculate the null space of $\mathcal{O}$.

Again, because of the complexity of symbolic matrix analysis, we run a Monte-Carlo simulation and considered the values below the calculation precision as zero. We highlight here, that this is strictly speaking not a mathematical proof. Indeed, theoretically, we might miss certain hyperplanes suggesting other unobservable modes. However, our findings correlate strongly with those in (Kelly and Sukhatme (2011)) and other work. Also, the results agree on the intuition. Moreover, the theory of (Hermann and Krener (1977)) states a proof of observability for a constant control input. Based on the vast number of simulation runs, the accordance to other work and the intuition, and the fact, that one constant control input is sufficient to prove locally weak observability we assume that, the hyperplanes found are correct. However, for certain specific motions, there might be more hyperplanes rendering different states unobservable. Our Monte-Carlo simulation shows that, for a generic motion with the above mentioned limitations, the following states X_U are jointly observable and only in one single dimension in this subspace unobservable

$$X_U = {}_x p_w^i,\ {}_y p_w^i,\ {}_x v_w^i,\ {}_y v_w^i,\ q_w^i,\ q_v^w \tag{3.56}$$

where ${}_x p_w^i,\ {}_y p_w^i,\ {}_x v_w^i,\ {}_y v_w^i$ are the x and y components of p_w^i and v_w^i respectively. We see that the visual scale λ, all bias terms b_w, b_a and even the full inter-sensor calibration p_i^c, q_i^c seems to be still observable. Intuitively, one would argue that the system has no information for its absolute yaw. This assumption is strongly supported by the fact that the z-component of p_w^i and v_w^i is observable but are only jointly observable in the x- and y-components.

From the rank test we know that one observation orthogonal to the rest is sufficient to reach full observability. As we consider the translational drift p_v^w later in Section 3.4.3, we discard here the possibility of measuring a component of p_w^i or v_w^i. This narrows the investigation for an additional measurement down to the components of q_w^i and q_v^w. Since there is rarely a practical case to measure a single element of a quaternion, we investigate the measurement of the roll, pitch or yaw angle of each quaternion. We list the

measurement h_6 for roll of q_w^i, h_7 for pitch of q_w^i, h_8 for yaw of q_w^i and h_9 to h_{11} for roll, pitch, yaw of q_v^w respectively:

$$h_6 = \arctan\left(\frac{2(q^i_{w_1} * q^i_{w_2} + q^i_{w_3} * q^i_{w_4})}{1 - 2({q^i_{w_2}}^2 + {q^i_{w_3}}^2)}\right) \tag{3.57}$$

$$h_7 = \arcsin\left(2(q^i_{w_1} * q^i_{w_3} - q^i_{w_2} * q^i_{w_4})\right) \tag{3.58}$$

$$h_8 = \arctan\left(\frac{2(q^i_{w_1} * q^i_{w_4} + q^i_{w_2} * q^i_{w_3})}{1 - 2({q^i_{w_3}}^2 + {q^i_{w_4}}^2)}\right) \tag{3.59}$$

$$h_9 = \arctan\left(\frac{2(q^w_{v_1} * q^w_{v_2} + q^w_{v_3} * q^w_{v_4})}{1 - 2({q^w_{v_2}}^2 + {q^w_{v_3}}^2)}\right) \tag{3.60}$$

$$h_{10} = \arcsin\left(2(q^w_{v_1} * q^w_{v_3} - q^w_{v_2} * q^w_{v_4})\right) \tag{3.61}$$

$$h_{11} = \arctan\left(\frac{2(q^w_{v_1} * q^w_{v_4} + q^w_{v_2} * q^w_{v_3})}{1 - 2({q^w_{v_3}}^2 + {q^w_{v_4}}^2)}\right), \tag{3.62}$$

with $q^i_{w_1}$ and $q^w_{v_1}$ being the real part of the quaternions q_w^i and q_v^w. We apply only one measurement at a time. This adds one of the following rows to $\mathcal{O}$:

$$\nabla L^0 h_6 = [0_{1\times 6} \quad \frac{\partial L^0 h_6}{\partial q_w^i} \quad 0_{1\times 18}] \tag{3.63}$$

$$\nabla L^0 h_7 = [0_{1\times 6} \quad \frac{\partial L^0 h_7}{\partial q_w^i} \quad 0_{1\times 18}] \tag{3.64}$$

$$\nabla L^0 h_8 = [0_{1\times 6} \quad \frac{\partial L^0 h_8}{\partial q_w^i} \quad 0_{1\times 18}] \tag{3.65}$$

$$\nabla L^0 h_9 = [0_{1\times 24} \quad \frac{\partial L^0 h_9}{\partial q_v^w}] \tag{3.66}$$

$$\nabla L^0 h_{10} = [0_{1\times 24} \quad \frac{\partial L^0 h_{10}}{\partial q_v^w}] \tag{3.67}$$

$$\nabla L^0 h_{11} = [0_{1\times 24} \quad \frac{\partial L^0 h_{11}}{\partial q_v^w}] \tag{3.68}$$

A rank test with Mathematica reveals full column rank of $\mathcal{O}$ for any additional measurement h_8 to h_{11}. However, h_6 and h_7 do not add a rank to $\mathcal{O}$. Mathematically, h_8 to h_{11} have all an orthogonal component to the other measurements h_1 to h_5. Thus, the gradients of their Lie derivatives have

a parallel component to the nullspace of $\mathcal{O}$ or, equivalently, an orthogonal component to the image space of $\mathcal{O}$. Conversely, h_6 and h_7 are fully parallel to the image space of $\mathcal{O}$ or, equivalently, fully orthogonal to the nullspace of $\mathcal{O}$ and, hence, do not add a rank to the system. This means that

- measuring the roll or pitch of the IMU with respect to to the world frame W does not add any new information.
- in contrast, measuring the roll or pitch drift between the vision frame V and world frame W yields a fully observable system. (This applies only if roll and pitch drifts are non-zero, otherwise, of course, the system still contains a yaw ambiguity).
- also measuring either yaw of the IMU with respect to the world frame W or measuring the yaw drift between the world frame W and vision frame V renders the system observable.

Note that the findings of Kelly in (Kelly and Sukhatme (2011)) about the IMU acceleration a and the angular velocity ω also apply on this system. In this dissertation, we do not focus in detail on the need of specific system excitements. We assume the flying MAV to have sufficient motion at any point in time.

We still consider the position drift p_v^w to be zero. Adding this 3D state without adding new measurements concerning the translational states simply results in 3 additional jointly observable variables with three additional unobservable directions. More precisely, the following states will only be jointly observable:

$$X_U = p_w^i,\ v_w^i,\ q_w^i,\ q_v^w,\ p_v^w \tag{3.69}$$

In addition to the unobservable mode spanning one dimension in the states mentioned in (3.56), we find three additional unobservable modes: each one spans one dimension in the subspaces $\{{}_x p_w^i,\ {}_x p_v^w\}$, $\{{}_y p_w^i,\ {}_y p_v^w\}$ and $\{{}_z p_w^i,\ {}_z p_v^w\}$ respectively. This means, that, in this setup, the position and the position drift are only jointly observable in each axis. We will consider position drift states p_v^w in Section 3.4.3 where we add a position sensor to the system. In the following we show how to tackle the unobservable states either by adding additional constraints on the vision system or by adding additional sensors.

In summary, we analyzed the observability of adding a 6DoF (possibly scaled) pose sensor – i.e. the camera – as a measurement sensor to our IMU based propagation model. We identified the unobservable modes and their inter-connectivity with in the state space by the following steps:

1. Construct the observability matrix $\mathcal{O}$ using Lie differentiation on the (unchangeing) propagation equations (3.54) and the (sensor specific) measurement equations (3.55)

2. On rank deficiency of $\mathcal{O}$, calculate its nullspace $\mathcal{O}_{ns}$. The columns of $\mathcal{O}_{ns}$ are the unobservable modes of the system (i.e. its symmetries). The system needs as many additional measurements with orthogonal components to each other and to the existing ones as the column rank of $\mathcal{O}_{ns}$ is. then, the system is fully observable.

3. Additional measurements (i.e. sensors) affecting these symmetries add corresponding rows to $\mathcal{O}$. If these rows are linearly dependent on the existing rows (rank of $\mathcal{O}_{ns}$ remains unchanged), the additional value measured was observable before. Otherwise, an unobservable value has been identified. We used this to identify that the global roll and pitch angle of the MAV are observable but the global yaw is not. Furthermore, roll, pitch, yaw of the vision-world drift are jointly observable and measuring only one of the three values renders the whole system observable.

We use this recipe to analyze in the following the addition of further sensors to account for the visual drifts. Note that the addition of sensors only adds different measurement equations. The propagation part remains unchanged (up to constant propagation of possibly added inter-sensor calibration states) since we continue using the IMU as propagation sensor.

A Note on the Rigid Body Assumption

In the whole system, we only model a rigid body but do not force it by measurements. Only the uncertainty of inter-sensor transformation does not increase with time. This allows the filter to adjust initialization errors as well as small transformation changes during flight. We could add the rigid body constraint as fix measurements. This results in the possibility to directly estimate the attitude q_i^c between the two sensors. This eliminates 4 States (q_i^c) from the rank deficient matrix discussed above. However, despite this additional constraint the matrix will still be rank deficient, since the rank deficiency does not rise from q_i^c. Using the rigid body constraint as a direct measurement is thus redundant.

3.4 Additional Sensors for Drift Elimination

In this section, we focus on additional sensors or properties of the vision algorithm in order to have a fully observable system. In particular, we present four base systems, each representing the availability of a certain kind of sensors. Even though we present the systems by using the most common sensor types of their class, the approaches are only limited to the type of the information the sensors yield, not to the sensors as such. Note that, because of the lack of any non-drifting position information, we only consider position drift in the third scenario where we add a global position sensor (section 3.4.3).

3.4.1 1DoF Drift Free Vision Box: Visual Compass

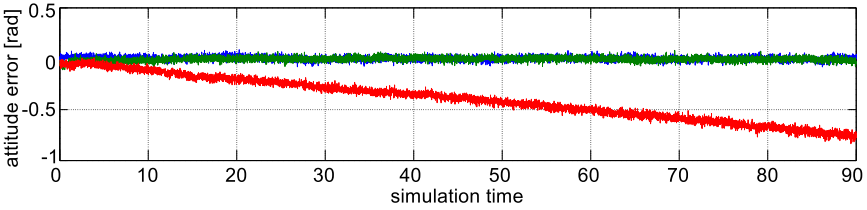
We first consider a vision algorithm not drifting in one angular direction. In open space, this is achieved using a visual compass. Points at infinity can be tracked to obtain a fix yaw direction. If we use such a visual algorithm we can add (3.68) to $\mathcal{O}$. We showed in section 3.3.2 that this renders the system observable. Based on this example it is immediate to see which row is added to $\mathcal{O}$ if other angular constrains are given by the vision system.

Second, we consider a task which is robust with respect to drift in a particular angular direction. An aerial vehicle is very sensitive to wrong roll and pitch estimates, however, when navigating locally, the yaw drift is less of importance. We can thus enforce the visual yaw drift to a constant value and again add (3.68) to $\mathcal{O}$. This, however, lets our world frame W drift along yaw together with the vision frame V. We have in essence the same setup as in (Kelly and Sukhatme (2011)) where the authors estimate the robot pose directly with respect to the visual frame V. Since they estimate gravity in V their system only drifts in yaw.

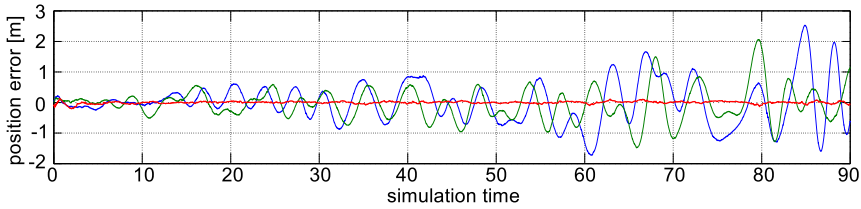
To summarize, we can either use specific vision algorithms which ensure no drift in one angular direction, or adapt our robot task such that it is insensitive to drift in one angular direction. However, we still have the position drift p_v^w. That is, our world frame W will still drift in position along with the vision frame V. In Fig. 3.6, we show the effects of the angular vision drift on the attitude, position and velocity estimates q_w^i, p_w^i and v_w^i respectively. Note that the filter still works consistently and correctly, however, the estimates differ from the values with respect to a truly fixed world frame.

3.4.2 Fix World Direction: Three Axis Magnetometer

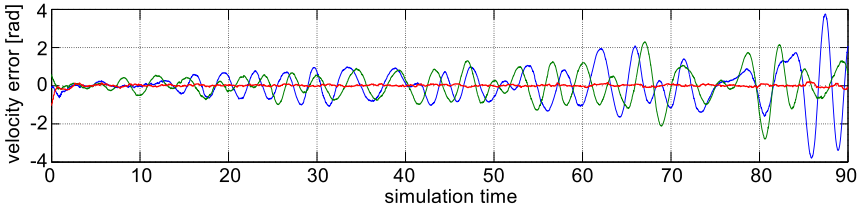
In this part we show how our core-sensor suite discussed in Section 3.1.1 and analyzed in Section 3.3 can be augmented with further sensors. We stress



(a)



(b)



(c)

Figure 3.6: Filter simulation on a robot task which is not sensitive to yaw drift. Even though the vision part drifts in yaw, we enforce the yaw offset between world frame W and vision frame V to be constant. The filter still works properly, but the world reference frame W is not a fixed inertial world frame anymore but drifting along yaw together with the vision frame V. We compare the filter values to ground truth in a truly fix world frame. In (a) the filter estimate on q_w^i directly reflects the yaw drift of the vision part. Of course this leads to different x- and y-estimates in position (b) and velocity (c). The rising amplitude does not mean a filter divergence, but reflects the growing yaw offset of W with respect to a truly fixed world frame.

here again that thanks to the introduction of a world reference frame W, we have a solid basis to easily include further sensors and to have the robot pose and covariance directly in the navigation frame.

We still assume a translational drift free visual part (or equivalently we let our world reference frame W drift in position together with the vision). In order to reach full observability of our core-sensor suite we include an absolute direction measurement with an orthogonal component to the gravity vector. Some possible sensors and their measurements can be:

- Sun sensor measuring the direction vector to the sun
- Second camera measuring the direction vector to a distinctive point at infinity (i.e. on the horizon as visual compass (Sturm and Visser (2009)))
- Magnetometer measuring the direction of the earth's magnetic field
- etc

A direction vector is defined in 3D by two angles. Further, a sensor which measures a direction has only to be calibrated in its attitude with respect to the robot (i.e. IMU in our case). The translational calibration parameters are irrelevant. We show that we can estimate the full attitude calibration parameters and the elevation angle of the direction vector in the world frame. Moreover, if this vector remains constant over time we are allowed to set the azimuth angle arbitrarily without loss of generality. If we use as an example the sun sensor, we only need a sun model describing its azimuth over the time of day. More interestingly, we do not need any model neither for the visual compass nor for the magnetometer - given that the robot's large scale interaction environment is still sufficiently small that it can be approximated by a plane with constant north.

We analyze the new system on the example of the Magnetometer. The replacement with other sensors of the same family as mentioned in the listing is immediate. The core-sensor suite is depicted in Fig. 3.3. Adding a direction measuring sensor yields the setup in Fig. 3.7. As discussed, we add an orientation q_i^m between the new sensor and the IMU as calibration parameter. Furthermore, we include the elevation angle α of the measured direction vector in our world frame W. Intuitively, it is clear that we can observe all 3DoF of the robot's attitude if we include a magnetometer. However, we show that we still have a self-calibrating system and do not need any model of the earth's magnetic field.

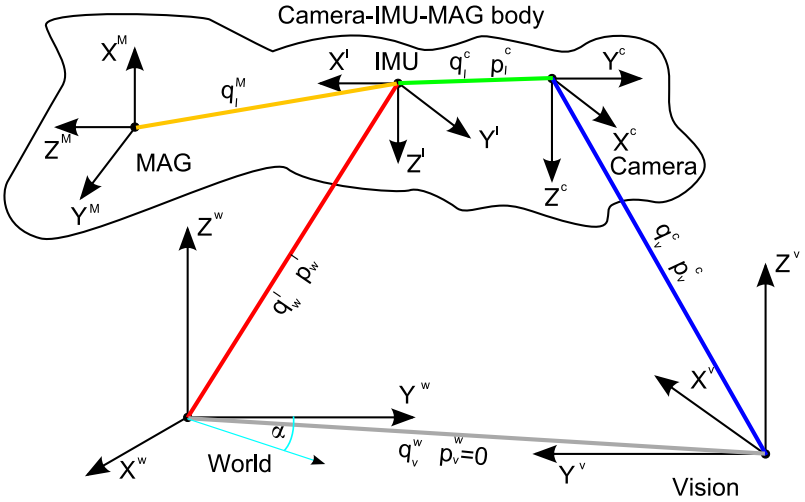


Figure 3.7: Visualization of the different coordinate frames in the setup including a direction measuring sensor. In this case we add a magnetometer. The setup is the same as in Fig. 3.3. However, we add as states the rotation between the magnetometer and the IMU q_i^m. Note that because of the directional nature of the measurement, we do not need a calibration state defining the translation with respect to the IMU. We also add the elevation angle α of the direction vector in the world frame.

If we include an additional sensor (world direction measurement), the filter design discussed in Section 3.1.1 only changes slightly. Since both states q_i^m and α do not increase their uncertainty over time, we model them noise-free like q_i^c and p_i^c in equations (3.19) and (3.24):

$$\dot{q}_i^c = 0 \quad \dot{\alpha} = 0 \tag{3.70}$$

$$\hat{\dot{q}}_i^c = 0 \quad \hat{\dot{\alpha}} = 0 \tag{3.71}$$

Thus, the state propagation matrices F_d and Q_c derived in Section 3.1.1 are padded with zeros in the newly added 4 state dimensions. This is a trivial change.

As measurement we compare the measured direction vector with the constant one in our world frame W

$$\begin{aligned} h_{12} &= C_{(q_i^m)} C_{(q_w^i)} m(\alpha) + n_m \\ h_{13} &= q_i^{mT} q_i^m, \end{aligned} \tag{3.72}$$

where $m(\alpha) = [0 \ \cos(\alpha) \ \sin(\alpha)]^T$ is the magnetic field direction vector with elevation angle α. And n_m is the zero mean and white Gaussian measurement noise of the magnetometer. The measurement h_{13} simply defines the unit norm of the quaternion q_i^m representing the rotation between magnetometer and IMU. Since we consider the magnetic field to be constant over the operation time and space of the robot, we are free to chose the azimuth angle of the direction vector. In this case, we chose our "north" to be along the positive y-axis of our world frame W. We apply the same method as used in Section 3.3 and obtain the following observability matrix $\mathcal{O}_{\mathcal{M}}$

$$\begin{bmatrix} \nabla L^0 h_1 \\ \nabla L^0 h_2 \\ \nabla L^0 h_{12} \\ \nabla L^0 h_3 \\ \nabla L^0 h_4 \\ \nabla L^0 h_5 \\ \nabla L^0 h_{13} \\ \nabla L^1_{f_0} h_1 \\ \nabla L^1_{f_0} h_2 \\ \nabla L^1_{f_0} h_{12} \\ \nabla L^1_{f_1} h_1 \\ \nabla L^1_{f_1} h_2 \\ \nabla L^1_{f_1} h_{12} \\ \nabla L^2_{f_0 f_0} h_1 \\ \nabla L_{f_0 f_2} h_1 \\ \nabla L_{f_0 f_0 f_2} h_1 \end{bmatrix} = \begin{bmatrix} \overbrace{}^{p_w^i} & \overbrace{}^{v_w^i} & \overbrace{}^{q_w^i} & \overbrace{}^{b_\omega} & \overbrace{}^{b_a} \\ C_v^{w\mathrm{T}} \lambda & 0 & [b_{1,3}] & 0 & 0 \\ 0 & 0 & [b_{2,3}] & 0 & 0 \\ 0 & 0 & [b_{3,3}] & 0 & 0 \\ 0 & 0 & 2 q_w^{i\mathrm{T}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & C_v^{w\mathrm{T}} \lambda & [b_{8,3}] & [b_{8,4}] & 0 \\ 0 & 0 & [b_{9,3}] & [b_{9,4}] & 0 \\ 0 & 0 & [b_{10,3}] & [b_{10,4}] & 0 \\ 0 & 0 & [b_{11,3}] & 0 & 0 \\ 0 & 0 & [b_{12,3}] & 0 & 0 \\ 0 & 0 & [b_{13,3}] & 0 & 0 \\ 0 & 0 & [b_{14,3}] & [b_{14,4}] & [b_{14,5}] \\ 0 & 0 & [b_{15,3}] & 0 & 0 \\ 0 & 0 & [b_{16,3}] & [b_{16,4}] & [b_{16,5}] \end{bmatrix}$$

$$
\begin{bmatrix}
[b_{1,6}] & [b_{1,7}] & 0 & [b_{1,9}] & 0 & 0 \\
0 & 0 & [b_{2,8}] & [b_{2,9}] & 0 & 0 \\
0 & 0 & 0 & 0 & [b_{3,10}] & [b_{3,11}] \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2q_i^{c\mathrm{T}} & 0 & 0 & 0 \\
0 & 0 & 0 & 2q_v^{w\mathrm{T}} & 0 & 0 \\
0 & 0 & 0 & 0 & 2q_i^{m\mathrm{T}} & 0 \\
\left[b_{8,6}\right] & [b_{8,7}] & 0 & [b_{8,9}] & 0 & 0 \\
0 & 0 & [b_{9,8}] & [b_{9,9}] & 0 & 0 \\
0 & 0 & 0 & 0 & [b_{10,10}] & [b_{10,11}] \\
\left[b_{11,6}\right] & [b_{11,7}] & 0 & [b_{11,9}] & 0 & 0 \\
0 & 0 & [b_{12,8}] & [b_{12,9}] & 0 & 0 \\
0 & 0 & 0 & 0 & [b_{13,10}] & [b_{13,11}] \\
\left[b_{14,6}\right] & [b_{14,7}] & 0 & [b_{14,9}] & 0 & 0 \\
\left[b_{15,6}\right] & 0 & 0 & [b_{15,9}] & 0 & 0 \\
\left[b_{16,6}\right] & [b_{16,7}] & 0 & [b_{16,9}] & 0 & 0
\end{bmatrix}
\begin{matrix}
\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
\end{matrix}
\tag{3.73}
$$

λ p_i^c q_i^c q_v^w q_i^m α

using the variables $b_{i,j}$ for better legibility. We use Mathematica to prove column full rank of $\mathcal{O}_{\mathcal{M}}$. This means that the system is locally weakly observable in all its states. Note that we used the unitary direction vector of the earth's magnetic field as measurement. Since the measurement equation h_{12} only contains rotations, the magnitude of the direction vector would also be observable. We could thus add a state ν defined as the scaling factor of $m(\alpha) = \nu[0 \ \cos(\alpha) \ \sin(\alpha)]^T$. The addition is seamless as the factor is considered noise free in the propagation:

$$\dot{\nu} = 0 \quad \hat{\dot{\nu}} = 0 \tag{3.74}$$

However, for other sensors such as the sun sensor or a visual compass, ν has no meaning. We keep the model general and consider thus a unit vector as measurement of the magnetometer.

In the above setup, we do not include bias terms on the direction sensor because there are several hardware implementations of direction sensors which do not suffer from this issue. Nevertheless, we briefly sketch the changes that apply when considering a magnetometer sensor with bias terms on each measured axis. The world magnetic field vector m would then be parametrized as $m = [m_x \ m_y \ m_z]^T$ and, using the magnetometer bias term b_m, the measurement functions in (3.72) will change to

$$h'_{12} = C_{(q_i^m)} C_{(q_w^i)} m + b_m + n_m \tag{3.75}$$

$$h_{13} = q_i^{mT} q_i^m, \tag{3.76}$$

We use the recipe discussed in Section 3.3.2 and see that the bias terms b_m are observable. Also, the z-component m_z of the magnetic field vector m is observable. Furthermore, one dimension of the subspace $\{m_x,\ m_y\}$ is observable (i.e. a symmetry spans only one dimension in this subspace). This is essentially equivalent to the observability of the elevation angle α and the need to fix an (arbitrary but constant) azimuth angle β. For example, setting the "task-north" along the x-axis of our world reference frame adds the measurement $m_y = 0$ and renders the whole system observable. This "task-north" may not be aligned with the magnetic north, but it is still a fix, non-drifting direction.

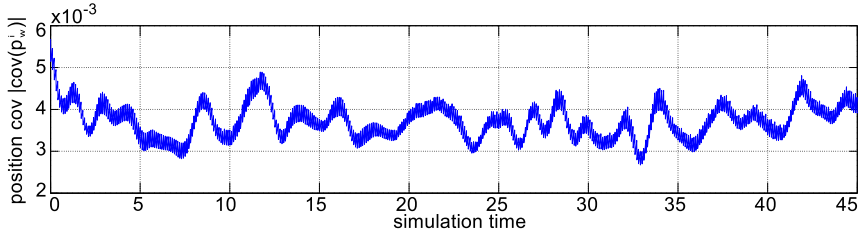
Even though this system is fully observable (if we use an arbitrary but fixed azimuth for the world reference frame) and self-calibrating, we still ignore the position drift p_v^w. Furthermore, since we have no correction on the position and not full certainty about the angular drift q_v^w, the uncertainty of the position p_w^i and velocity v_w^i increase with distance to the origin of the world reference frame W. It is very intuitive that a small uncertainty in q_v^w leads to a large uncertainty at positions far away from the origin of W. The strengths of our approach is that this is correctly reflected in the state covariance matrix. Whereas other approaches underestimate this uncertainty because they do not include the uncertainty of the angular drift. Our correct covariance estimate is for example highly useful for loop closure detections. However, this is not the focus of this paper. Figure 3.8 depicts a trajectory around the origin and one with increasing distance to the origin.

Note that the increasing uncertainty does not reflect the certainty of the position drift but only the certainty of the angular drift. If we would like to augment the pose uncertainty with the position drift we would need to add the states p_v^w to our system. However, we saw in section 3.3 that these states render the system unobservable. We tackle this issue in the following.

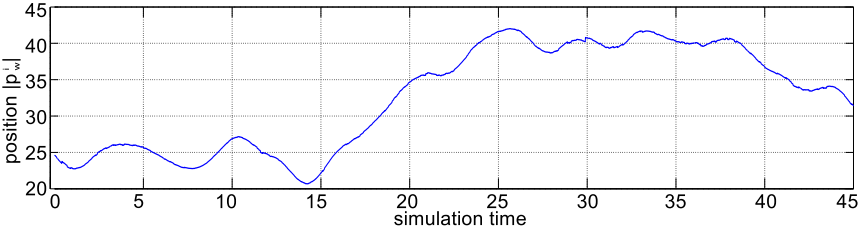
3.4.3 Absolute Position Measurement: GPS

In this part we consider additional sensors yielding an absolute (i.e. drift free) position measurement. Examples of such sensors are

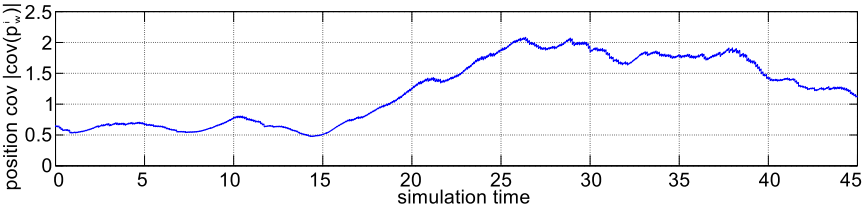
- 3DoF Laser tracking system (i.e. theodolite)
- Vision based tracking system (for example Vicon)
- GPS
- etc.



(a)



(b)



(c)

Figure 3.8: Filter simulation on a robot staying in the vicinity of the world frame origin (a) and navigating further away ((b) and c)). In (a) the norm of the distance p_w^i remains approximately the same. Thus the position covariance stays at a constant level after convergence. In (b) we plot the norm of p_w^i for a different run. At around sec 15 the norm increases and so does the covariance in (c). Due to the uncertainty in the attitude drift, we model correctly the uncertainty in the positions further away from the origin.

All these sensors have in common that they do not measure any rotation, but an absolute, drift free 3D position. As previously mentioned, our modular approach includes a world reference frame W. This frame is decoupled from the vision system and can thus be used as reference frame for the absolute position measurements of the above listed sensors.

As we can see in Fig. 3.9 the core-sensor suite is augmented by the additional sensor including its calibration parameters p_i^g denoting the distance between the IMU and the additional sensor. Also, the translational drifts of the vision system p_v^w are added to the state. As an example we consider a GPS sensor. However, any sensor with the same measurement type (i.e. absolute 3D position) is applicable. Note that the attitude of this sensor with respect to the IMU is irrelevant, since it only measures a position.

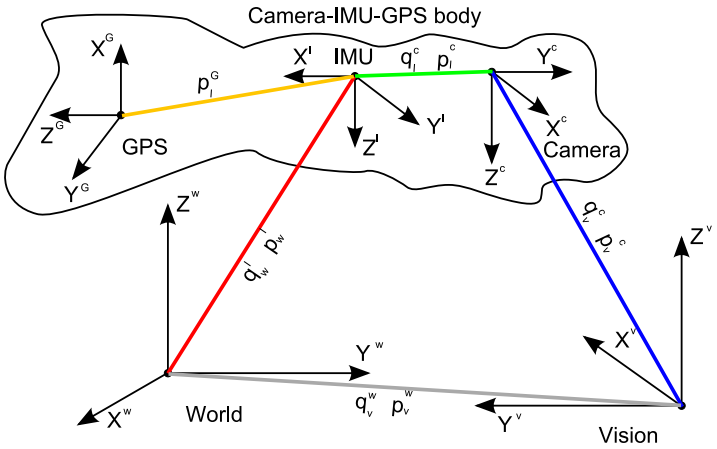


Figure 3.9: Visualization of the different coordinate frames in the setup including a position measuring sensor. In this case we add a GPS. The setup is the same as in Fig. 3.3. However, we add as states the translation between the GPS and the IMU p_i^g. Note that because of the translational nature of the measurement, we do not need a calibration state defining the rotation with respect to the IMU. We also add the position drift state p_v^w defining the translational drift of the vision frame V with respect to the world frame W.

Once again, the filter design discussed in Section 3.1.1 changes only slightly. The inter-sensor calibration state p_i^g does not increase its uncertainty over time. Thus, we model it noise-free like q_i^c in equations (3.19) and (3.24). Also, we model the drift state p_v^w as a spatially drifting state and do not model it as a temporal random walk. Hence, we have:

$$\dot{p}_i^g = 0 \quad \dot{p}_v^w = 0 \tag{3.77}$$
$$\hat{\dot{p}}_i^g = 0 \quad \hat{\dot{p}}_v^w = 0 \tag{3.78}$$

Therefor, the state propagation matrix F_d is padded as well with zeros for the newly added 6 state dimensions of p_i^g and p_v^w. The covariance matrix Q_c derived in Section 3.1.1 is also augmented with zero entries of dimension 6 for p_i^g and p_v^w.

As measurement, we compare the measured position with the estimate of the filter. Note that the camera position measurement changes since we added the translational drift state p_v^w

$$h_{14} = p_w^i + C_{(q_i^w)}^T + p_i^g + n_g \tag{3.79}$$
$$h_1' = (p_v^w + C_{(q_v^w)}^T(p_w^i + C_{(q_w^i)}^T p_i^c))\lambda \tag{3.80}$$

where n_g is the zero mean white Gaussian measurement noise of the position measurement sensor. We apply the same method as under Section 3.3 and obtain the following observability matrix $\mathcal{O}_\mathcal{G}$

$$\begin{bmatrix} \nabla L^0 h_1' \\ \nabla L^0 h_2 \\ \nabla L^0 h_{14} \\ \nabla L^0 h_3 \\ \nabla L^0 h_4 \\ \nabla L^0 h_5 \\ \nabla L^1_{f_0} h_1' \\ \nabla L^1_{f_0} h_2 \\ \nabla L^1_{f_0} h_{14} \\ \nabla L^1_{f_1} h_1' \\ \nabla L^1_{f_1} h_2 \\ \nabla L^1_{f_1} h_{14} \\ \nabla L^2_{f_0 f_0} h_1' \\ \nabla L_{f_0 f_2} h_1' \\ \nabla L_{f_0 f_0 f_2} h_1' \end{bmatrix} = \begin{bmatrix} \overbrace{p_w^i}^{p_w^i} & \overbrace{v_w^i}^{v_w^i} & \overbrace{q_w^i}^{q_w^i} & \overbrace{b_\omega}^{b_\omega} & \overbrace{b_a}^{b_a} \\ C_v^{w\mathrm{T}}\lambda & 0 & [c_{1,3}] & 0 & 0 \\ 0 & 0 & [c_{2,3}] & 0 & 0 \\ I_{d_3} & 0 & [c_{3,3}] & 0 & 0 \\ 0 & 0 & 2q_w^{i\mathrm{T}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & C_v^{w\mathrm{T}}\lambda & [c_{7,3}] & [c_{7,4}] & 0 \\ 0 & 0 & [c_{8,3}] & [c_{8,4}] & 0 \\ 0 & I_{d_3} & [c_{9,3}] & [c_{9,4}] & 0 \\ 0 & 0 & [c_{10,3}] & 0 & 0 \\ 0 & 0 & [c_{11,3}] & 0 & 0 \\ 0 & 0 & [c_{12,3}] & 0 & 0 \\ 0 & 0 & [c_{13,3}] & [c_{13,4}] & [c_{13,5}] \\ 0 & 0 & [c_{14,3}] & 0 & 0 \\ 0 & 0 & [c_{15,3}] & [c_{15,4}] & [c_{15,5}] \end{bmatrix}$$

$$
\begin{matrix}
\underbrace{\begin{bmatrix} c_{1,6} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \left[c_{7,6}\right] \\ 0 \\ 0 \\ \left[c_{10,6}\right] \\ 0 \\ 0 \\ \left[c_{13,6}\right] \\ \left[c_{14,6}\right] \\ \left[c_{15,6}\right] \end{bmatrix}}_{\lambda} &
\underbrace{\begin{bmatrix} c_{1,7} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ c_{7,7} \\ 0 \\ 0 \\ c_{10,7} \\ 0 \\ 0 \\ c_{13,7} \\ 0 \\ c_{15,7} \end{bmatrix}}_{p_i^c} &
\underbrace{\begin{bmatrix} 0 \\ c_{2,8} \\ 0 \\ 0 \\ 2q_i^{c\mathrm{T}} \\ 0 \\ 0 \\ c_{8,8} \\ 0 \\ 0 \\ c_{11,8} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{q_i^c} &
\underbrace{\begin{bmatrix} c_{1,9} \\ c_{2,9} \\ 0 \\ 0 \\ 0 \\ 2q_v^{w\mathrm{T}} \\ c_{7,9} \\ c_{8,9} \\ 0 \\ c_{10,9} \\ c_{11,9} \\ 0 \\ 0 \\ 0 \\ 0 \\ c_{15,9} \end{bmatrix}}_{q_v^w} &
\underbrace{\begin{bmatrix} {}^I d_3 \lambda \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{p_v^w} &
\underbrace{\begin{bmatrix} 0 \\ 0 \\ R_w^{iT} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ c_{9,11} \\ 0 \\ 0 \\ c_{12,11} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{p_i^g}
\end{matrix}
\tag{3.81}
$$

using the variables $c_{i,j}$ for better legibility. We use Mathematica to prove column full rank of $\mathcal{O}_\mathcal{G}$. This means that the system is locally weakly observable in all its states.

It is intuitively clear that an absolute position measurement with an IMU yields the full 6DoF robot pose. The camera in our core-sensor suite would not be needed. However, we note that global tracking systems are generally slow and/or very inaccurate and maybe not available over short periods of time. Thus the camera helps to bridge these deficiencies. Furthermore we showed with the observability analysis that we can estimate the inter-sensor parameters p_i^g as well. Note that we do not need a direction measurement, nor do we have the issue of rising uncertainties with increased distances towards then origin as we had in the example of the magnetometer.

Note About Pressure Sensors

Instead of a 3DoF position measurement we can also think of a single axis position sensor. It is, however, crucial that this measurement is drift free. Otherwise the system will not be observable, since the visual drift and the drift of the additional sensor are competing with each other. Thus, a pressure sensor that is prone to drift because of atmospheric changes will not add to the system's observability.

3.4.4 Combining Absolute Position and Fix World Direction

In the previous parts of this section, we discussed the addition of one single sensor to the core-sensor suite. Adding more sensors is straight forward but

needs attention to the three base scenarios discussed above. Without going into details, we here briefly discuss the addition of a direction measurement and an absolute position measurement. Again, as examples we consider the two sensors magnetometer and GPS.

For a constant direction vector in the world frame W, we saw in Section 3.4.2 that we can estimate its elevation angle in the filter and assume an arbitrary azimuth to which W will be oriented to. Furthermore, in Section 3.4.3 we saw that an absolute position measurement is sufficient to determine the translational and rotational drift as well as all inter-sensor parameters. The arbitrarily selected azimuth in Section 3.4.2 and the fully defined and observable system of Section 3.4.3 suggests an over-determination if both sensors are added to the core-sensor suite. In fact, with both sensors we can even estimate the azimuth angle of the measured direction vector in W. For the earth's magnetic field we modify $m(\alpha)$ from Section 3.4.2 to

$$m(\alpha, \beta) = [\sin(\beta)\cos(\alpha) \;\; \cos(\beta)\cos(\alpha) \;\; \sin(\alpha)]^T; \tag{3.82}$$

with β being the azimuth angle. It is intuitively clear that this system is fully observable and that the magnitude of the magnetic field vector could be estimated as well, as discussed in Section 3.4.2.

3.4.5 From Visual Pose to Visual Body-Velocity Sensor

So far, we added two sensor types to our core-sensor suite. The first type provided a direction measurement whereas the second provided a global position measurement. Furthermore we considered the camera as a scaled pose sensor drifting in 6DoF. The latter implies that we use the camera as a *map-based* sensor. Map-based visual pose estimators are prone to lose this map and thus to fail. In Chapter 2 we presented the camera as such a pose sensor but also as a body-velocity estimator which workes *map-free*. This map-free approach can be used to velocity control the MAV and is thus a viable method to initialize and bridge failures of the map-based approach. We analyze the observability of the states of such a system and show practical results in the next chapter.

The unified, scaled camera velocity $z_v = \eta\vec{v}$ recovered in section 2.2 is treated as a measurement to an EKF framework featuring a camera and an IMU. In the following, we describe very briefly the setup of the filter and then focus on its observability. This work is published in (Weiss et al. (2012b)).

Filter Update and Coupling with Optical Flow Measurements

In order to un-rotate the optical flow vectors we integrate the IMU's gyroscope readings between two camera frames and calculate the relative rotation using a first-order quaternion integration. It is important to note the following:

- The initialization for the gyroscope biases as filter states is accurate. In contrast to the accelerometer biases which interfere with the MAV's initial attitude, the gyroscope biases can directly be measured at the initial phase where the MAV stands still.
- The change of the gyroscope biases over time is tracked by the filter as a filter state – i.e. the quaternion integration with the gyroscope readings is done after statistically optimal bias compensation.
- Both devices IMU and camera are time synchronized.

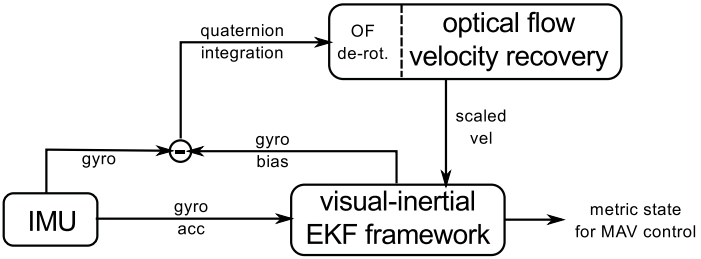


Figure 3.10: System setup for our semi-tightly coupling of inertial-optical flow based velocity recovery. The IMU is used as a prediction model in the EKF propagation phase, while the gyroscopes are used in a first-order quaternion integration to recover the relative rotation between two camera frames. The gyroscopic measurements are bias-compensated using the estimated bias term in the filter. The visual part un-rotates the optical flow measurements using the relative rotation such that we can apply (2.4) to determine the arbitrarily scaled camera velocity to be used in the EKF update. (Weiss et al. (2012b))

The whole framework, depicted in Fig. 3.10, is a mixture of sensor colligation (the IMU is used to un-rotate the optical flow vectors) and statistical fusion (EKF framework). The camera-velocity measurement can be described as:

$$z_v = (C_{(q_i^c)} C_{(q_w^i)} v_w^i + C_{(q_i^c)} (\lfloor \omega \rfloor p_i^c)) L + n_v \ , \tag{3.83}$$

Observability Analysis

The propagation equations are summarized in (3.54) in Section 3.3. Of course, in this setup, the drift states p_v^w and q_v^w between the vision and the world reference frame are superfluous since we directly recover the camera body-velocity. There is no map and hence no vision frame.

The measurement equations are:

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} (C_{(q_i^c)} C_{(q_w^i)} v_w^i + C_{(q_i^c)} (\lfloor \omega \rfloor p_i^c)) L \\ {q_w^i}^T q_w^i \\ {q_i^c}^T q_i^c \end{bmatrix}, \tag{3.84}$$

with h_2 and h_3 being the constraints of unit norm quaternions of rotation.

We denote the gradient with respect to the state variables of the zero order Lie derivative of h_n as $\nabla L^0 h_n$, and its first order derivative with respect to f_m as $\nabla L^1_{f_m} h_n$. Following the suggestions in (Kelly and Sukhatme (2011); Mirzaei and Roumeliotis (2008); Weiss and Siegwart (2011)), we obtain the observability matrix $\mathcal{O}$:

$$\mathcal{O} = \begin{bmatrix} \nabla {L^0}_{h_1} \\ \nabla {L^0}_{h_2} \\ \nabla {L^0}_{h_3} \\ \nabla {L^1_{f_0}}_{h_1} \\ \nabla {L^1_{f_1}}_{h_1} \\ \nabla {L^1_{f_2}}_{h_1} \\ \nabla {L^2_{f_0 f_0}}_{h_1} \\ \nabla {L^2_{f_1 f_0}}_{h_1} \\ \nabla {L^2_{f_2 f_0}}_{h_1} \end{bmatrix}.$$

A rank condition calculation reveals rank deficiency (rank 20 instead of 24). An analysis of the continuous symmetries as proposed in (Martinelli (2011)) reveals that the position states and the yaw state of the IMU attitude with respect to the world reference frame are not observable. This is not surprising, since we only have camera velocities as measurement and thus no information about the absolute position and yaw orientation. The absolute roll and pitch angle of the IMU with respect to the world frame is made observable by the IMU's gravity sensing. Moreover, the analysis shows, that not only the visual scale factor L is observable, but also all 6DoF of the inter-sensor calibration states between IMU and camera.

We recall that in Section 2.2, we discussed the two possibilities of propagating the arbitrary visual scale factor within the visual framework or use a scene-depth normalization rendering the scale factor only spatially and not temporaly drifting. When propagating the scale in the visual framework, we observed the issue of accumulated scale error degrading the velocity estimate in time. In contrast, using a scene-depth normalization approach

always yields the same unifying scale factor at a given scene depth. The only information needed are then two consecutive images.

Knowing that the visual scale is only spatially drifting and that the scale factor as such is observable in the filter framework, renders the filter a better candidate for the scale propagation than the visual framework itself. Note that, for a down-looking camera setup, the spatial scale drift only occurs on changes in altitude.

The fact that only the position p_w^i and yaw part of the MAV's attitude q_w^i is unobservable suggests that the MAV is still able to self-calibrate its sensor suite and perform metric velocity control. However, it will slowly drift in position and yaw direction. This is nevertheless sufficient for short-term stabilization of the MAV. We show in Section 4.2.7 that this is sufficient to initialize a more elaborate, map-based monocular pose estimator for full MAV control.

Towards inertial-optical flow based position control

Our focus in the inertial-optical flow approach is to have a robust visual algorithm not relying on the re-detection of certain features or any sort of feature history. However, we shortly highlight the possible extension of our inertial-optical flow velocity recovery approach in order to eliminate position drift.

If we assume known camera calibration and project the camera readings to the unit sphere, we can apply the intercept theorem. Therefor, the optical flow $\dot{x}$ is defined as

$$\dot{x} = \frac{v}{D} \sin \alpha \ , \tag{3.85}$$

where v is the camera-velocity vector, D is the distance to the feature and α is the angle between v and the direction to the feature. Given the camera model, we can measure α (and $\dot{x}$). Hence we can add an extra measurement per observed feature including its optical flow and angle α to the EKF framework discussed above:

$$h_i = \frac{\dot{x}}{\sin \alpha} = \frac{v}{D} \ . \tag{3.86}$$

For one such measurement $h_4 = \frac{v}{D}$ we can define the MAV position with respect to a single feature as $D = \sqrt{x^2 + y^2 + z^2}$. The gradient of its Lie derivative with respect to the state space yields $\nabla L^0 h_4$. For a general movement, one such measurement adds a 3×6 matrix block of rank 3 to the observability matrix $\mathcal{O}$. The entries of this matrix block are in particular non-zero at the indices of p_w^i since $D = D(p_w^i)$. This means, that (for general movement), the position of the MAV is observable and thus drift-free. In

essence, we have a hyperplane that still renders the position unobservable. This hyperplane is a sphere around one feature or a circle between two features. Naturally, this is the region where D remains unchanged. In general, and as we know it from the Perspective-3-Point algorithm, we eliminate all such hyperplanes by observing 3 or more features. This means, we would need to extend our approach with a feature history of 3 or more features.

3.4.6 Multi-Sensor System Summary

In the previous sections, we discussed specific sensor additions to our core-sensor suite. We classified each specific sensor to a certain type of sensor class and discussed the additional states in the system and their observability. In this section, we increase the level of abstraction and provide an overview of additions of individual measurements to our system. For example, the camera position measurement can be divided into 3 individual position measurements. Here, we only consider the IMU as fix part of our system and as propagation sensor. Thus, the propagation equations (3.54) remain the same for all additions. Additional zero order hold propagations may apply from additional measurement sensors as we discussed above. Since we are considering micro helicopters, we assume that the system is excited by a general motion (i.e. all axes in acceleration and angular velocities are excited). Further analysis on the excitation of specific axis can be performed simply by setting the corresponding inertial reading to zero. In general, this analysis does not provide information an the amount of excitation needed — this should be tackled in the theoretical analysis of the implemented, linearized and discretized system (Mourikis (June 2008)). Our overview contains the following sensor types

- 1D scaled world position measurement with a possible spatial position and attitude drift (for example the position from the camera as a pose sensor)
- 1D scaled world velocity measurement with possible spatial attitude drift (for example GPS velocity)
- world direction measurement without model (for example sun sensor, direction vector's elevation and azimith angle are in the filter's state-space)
- 1D body-velocity measurement (for example the camera as a scaled body-velocity sensor)

Fig. 3.12 summarizes all analyzed sensor types and their observable states in the system. We detail the majority of this figure's entries in the following paragraphs, however, we refer to Section 3.3 to get detailed insight about the procedure of the analysis. Note that we do not anymore list the measurements to enforce unit length of the quaternions but still include them for the observability analysis. Hence, a quaternion has a maximum of three unobservable states (i.e. roll, pitch, yaw).

1D World Position Measurement

This type of sensor measures one arbitrarily scaled coordinate with respect to its own reference frame. We can think of this as a one axis position measurement of the estimated camera pose. First, we investigate a measurement in x. From the propagation equations, the following states are involved p_w^i, v_w^i, q_w^i, b_w, and b_ω. We account for the scaled measurement (state λ) and the possible position drift ${}_xp_v^w$ in x and attitude drift q_v^w respectively. We have the 3D distance between the sensor and the IMU p_i^s as only inter-sensor calibration state. The attitude between sensor and IMU is irrelevant when measuring the vehicle's position. This yields the following 25-element state vector

$$X_{S_x} = [p_w^i \; v_w^i \; q_w^i \; b_\omega \; b_a \; \lambda \; {}_xp_v^w \; q_v^w \; p_i^s]^T \tag{3.87}$$

for the observability analysis we use the propagation equations of 3.54 and the following measurement

$$h_{S_x} = ({}_xp_v^w + {}_x\,[C^T_{(q_v^w)}(p_w^i + C^T_{(q_w^i)}p_i^s)])\lambda \tag{3.88}$$

where ${}_x[v]$ denotes the first element (i.e. the x-part) of the vector v.

We perform the observability analysis as we did in Section 3.3.2 and summarize the results in the first three lines of Fig. 3.12. For a general motion and non-zero state values the corresponding observability matrix has rank 17 (of 25). There are no symmetries along b_ω, b_a, λ, and p_i^s. This means that the scale as well as the (inter-)sensor calibration states are directly observable by only one 1D position sensor. If we analyze the symmetries of the system, we see that the remaining states are jointly observable. More precisely, a 5-dimensional subspace of the 13-dimensional space of jointly observable states is observable. One of the 8 unobservable dimensions (J1 in Fig. 3.12) is only spanned by p_w^i and ${}_xp_v^w$ reflecting the intuitive joint observability of global position drift ${}_xp_v^w$ and global position p_w^i. The remaining 7 unobservable dimensions are spanned by p_w^i, v_w^i, q_w^i, q_v^w. The result is identical for sensors measuring the y or z direction (using ${}_yp_v^w$, ${}_zp_v^w$ and ${}_y[\cdot\,]$, ${}_z[\cdot\,]$ accordingly).

The effect of concatenating these three orthogonal sensors is listed in the second and third line of Fig. 3.12.

We can now improve our imaginary sensor and think of such a sensor not drifting in attitude. This is for example represented by a pressure sensor (measuring the z axis with drift and possible scale). $C^T_{(q^w_v)}$ in (3.88) is then a fix unit rotation and q^w_v is not in the state space anymore. The analysis of the continuous symmetries yields the following 6 unobservable modes (we again consider a sensor measuring the x axis):

- 4 unobservable modes span each the subspace ${}_yp^i_w$, ${}_zp^i_w$, ${}_yv^i_w$, and ${}_zv^i_w$ respectively. This was expected, since, without having an attitude drift in the measurement, we have no information on the global position or velocity in y and z. Since each mode spans one state dimension and they have no inter-connectivity, we call them directly unobservable and they can be omitted in the state space and the system.

- 1 unobservable mode spans a subspace space of ${}_xp^i_w$, ${}_xp^w_v$. Again, this is not surprising and means that the absolute position in x is only jointly observable with the drift of the measurement in this axis.

- 1 unobservable mode spans a subspace of the vehicle's attitude q^i_w. This means, that we cannot determine the attitude of our MAV with only this sensor mounted, even if it would not drift at all.

Excluding ${}_yp^i_w$, ${}_zp^i_w$, ${}_yv^i_w$, and ${}_zv^i_w$ from the system yields a system of 17 states. The rank test shows a rank of 15 and the test on continuous symmetries shows the two symmetries corresponding to the dimension of the rank deficiency span each a subspace of $[{}_xp^i_w, {}_xp^w_v]$ and q^i_w respectively. In fact, when considering a sensor measuring a drifting position in z (e.g. pressure sensor) the latter symmetry only spans the global yaw. These findings are listed in lines four and five in Fig. 3.12.

We can summarize the properties of 1D drifting, scaled position sensor with rotational drift:

- The inter-sensor calibration states p^s_i as well as the scale λ and IMU biases b_ω and b_a are directly observable. No prior calibration procedures are needed.

- If the sensor has no rotational drift, the velocity in the specific direction becomes observable. Also, the analysis on the symmetries show that the MAV attitude and 1D position are decoupled: either measuring the position or the position drift renders the other state observable

and measuring one element (roll, pitch or yaw) of the attitude renders the whole attitude observable. Specifically, if we add a sensor which yields perpendicular measurements to the existing (e.g. adding a sensor measuring the y position with drift and scale) renders the whole attitude observable whereas the position and position drift states keep being per sensor only jointly observable.

- We obtain a roll and pitch stable system in the special case of a sensor measuring the z position with drift. In practice, we can stabilize an MAV with a pressure sensor only. It will drift in x, y and yaw according to the IMU integration-error and in z according to the position drift of the sensor. Roll and pitch stability, however, prevent the MAV to crash.

1D World Velocity Measurement

This type of sensor measures one arbitrarily scaled velocity in one axis with respect to its own reference frame. Since we measure a first order spatial information (velocity) we only need to consider a potential rotational drift q_v^w. The state space is

$$X_{S_{vx}} = [v_w^i \; q_w^i \; b_\omega \; b_a \; \lambda \; q_v^w \; p_i^s]^T \tag{3.89}$$

for the observability analysis we use the propagation equations of 3.54 and the following measurement

$$h_{S_{vx}} =_x [C^T_{(q_v^w)}(v_w^i + C^T_{(q_w^i)} \lfloor \omega_\times \rfloor p_i^s)])\lambda \tag{3.90}$$

For a general motion and non-zero state values the corresponding observability analysis has rank 16 (of 21). The analysis of continuous symmetries shows that, for this sensor type as well, all inter sensor calibration states b_w, b_ω, p_i^s and the scale factor λ are directly observable even though the sensor only measures one axis. The states v_w^i, q_w^i, q_v^w span 5 symmetries in theis 9-dimensional state space (i.e. only 4 of the remaining 9 dimensions are observable). In Fig. 3.12 we show this finding in line 6 and summarize the effect of concatenating this type of sensor in different axes with orthogonal components in lines 7 and 8.

If we again improve our sensor to have no attitude drift, we can reduce the state space according to the above discussion on the 1D position sensor. This omits q_v^w as well as the remaining two axes in velocity ${}_y v_w^i$, and ${}_z v_w^i$. The remaining state space is of dimension 15 with an observability matrix of rank 14. Since the velocity the sensor measures becomes observable when

we do not have rotational drift, we have only one unobservable dimension in the attitude. A single additional observation (roll, pitch, or yaw) renders the whole attitude observable. In fact, in the special case when the sensor is measuring the velocity in z only the global yaw remains unobservable and the system is stable in roll and pitch. This is summarized in lines 9 and 10 of Fig. 3.12.

World Direction Measurement

This type of sensor measures a direction vector in the world frame. Practical examples may be a magnetometer, a sun sensor or a visual approach tracking a feature on the horizon (visual compass). Since the direction is of angular nature, we need a calibration state defining the rotational offset between the sensor and the IMU q_i^s. The direction vector in the world frame can be defined by an elevation angle state α and an azimuth angle state β. Because of the angular nature, we only consider the IMU gyro bias b_ω and the IMU (or MAV) attitude q_w^i. We omit all translational states p_w^i, v_w^i, b_a, λ, p_i^s. The state space is

$$X_{S_{vx}} = [q_w^i \; b_\omega \; q_v^w \; q_i^s \; \alpha \; \beta]^T \tag{3.91}$$

for the observability analysis we use the propagation equations of 3.54 and the following measurement

$$h_{S_{dir}} = C_{(q_i^s)} C_{(q_w^i)} C_{(q_v^w)} m(\alpha, \beta) \tag{3.92}$$

with $m(\alpha, \beta) = [\sin(\beta)\cos(\alpha) \;\; \cos(\beta)\cos(\alpha) \;\; \sin(\alpha)]^T$. For a general motion and non-zero state values the corresponding observability analysis yields rank 11 (of 17). Even though we assume a world drift of the direction vector, the (inter-)sensor calibration states b_ω and q_i^s are still observable. That is, a sun sensor is capable of calibrating the sensor suite even without having a sun direction model. 5 symmetries span a subspace of the states q_w^i, q_v^w, α and β and one additional symmetry spans a subspace of q_w^i. That is, 2 of these 8 joint dimensions are observable.

When we assume a non-drifting world vector (magnetic field) q_v^w, becomes a constant unit rotation and can be neglected in the state space. The observability analysis yields then rank 10 (of 13) and essentially reduces the unobservable modes by q_v^w. That is, two symmetries span a subspace of q_w^i, α and β and an additional symmetry spans a subspace of q_w^i such that 2 of these 5 joint dimensions are observable.

This implies that, even if we had a model of the world direction vector and can determine α and β we still have one unobservable dimension in the

global attitude. Measuring one single element orthogonal to this symmetry renders then the global attitude observable. These findings are summarized in Fig. 3.12 in lines 11 and 13. Line 12 in the same figure shows the case of an attitude sensor measuring the attitude in its own reference frame(for example the camera). The results are similar except that the world vector's elevation and azimuth angles α and β do not apply, eliminating 2 dimensions of the system.

1D Body-Velocity Sensor

This type of sensor measures a possible scaled body-velocity in one axis. A three axes body-velocity sensor might be a camera using our proposed inertial-optical flow approach (Section 2.2). For this sensor type we need to introduce both, translational p_i^s and rotational q_i^s inter-sensor calibration states. Because the measurement is possibly scaled, we use the scale state λ as well. Furthermore, we use all states related to the propagation step except the global position since we measure velocities only.

The state space is

$$X_{S_{bx}} = [v_w^i \; q_w^i \; b_\omega \; b_a \; \lambda \; q_i^s \; p_i^s]^T \tag{3.93}$$

for the observability analysis we use the propagation equations of 3.54 and the following measurement

$$h_{S_{dir}} = (C_{(q_i^s)} C_{(q_w^i)} v_w^i + C_{(q_i^s)} (\lfloor \omega \rfloor p_i^s))\lambda \tag{3.94}$$

For a general motion and non-zero state values the corresponding observability analysis yields rank 17 (of 21). From the analysis of the continuous symmetries (i.e. nullspace of $\mathcal{O}$), we see that the scale factor λ and the IMU calibration states b_ω and b_a are directly observable. So are the MAV states ${}_z v_w^i$, roll and pitch and we see that with a body-velocity measurement along one axis is sufficient to recover roll and pitch of the MAV. However, not all inter-sensor calibration states are observable. With the analysis of the continuous symmetries we see that we have one symmetry spanning a subspace in ${}_x v_w^i$, ${}_y v_w^i$, and ${}_{yaw} q_w^i$. This reflects the evident non-observability of global yaw. More interesting are the two symmetries spanning a subspace of p_i^s and one symmetry spanning a subspace of q_i^s. We reflect this fact in line 14 of Fig. 3.12.

From this analysis we see that the global yaw is an inherent unobservable mode whereas the inter-sensor calibration seems to be on a per-sensor basis. Furthermore, we only have one symmetry in the inter-sensor attitude

direction. This reflects the fact that the attitude is fully defined by two directional vectors (one direction vector defines to angles of rotation). Adding a second sensor measuring an orthogonal body-velocity component with respect to the first one renders thus the inter-sensor attitude q_i^s observable and reduces one symmetry in the translational inter-sensor state p_i^s. Measuring all three body-velocity axis as we do in our approach discussed in Section 2 renders the full calibration of the sensor suite observable. This example is shown in line 17 of Fig. 3.12.

Concatenation of Multiple Sensor Types

From the previous analysis we saw that in most cases (except the 1D body-velocity sensor) we are able to determine the full inter-sensor calibration including the biases of the IMU and a possible scale factor. This leads to the conclusion that tedious prior calibration procedures can be avoided. We also see that rarely the full MAV pose can be estimated with a 1D sensor in order to control the MAV appropriately. To mitigate this issue we look at the concatenation of different sensor types. For example, our inertial-optical flow approach in Section 2.2 represents three concatenated 1D body-velocity sensors. With the analysis above we can directly predict the observable states and unobservable modes.

In the following we show the idea behind the concatenation and explain which gains we can expect from concatenating multiple sensors.

Let the non-linear System Σ be

$$\begin{aligned} \dot{x} &= f(\vec{x}, \vec{u}) \\ \vec{z} &= \vec{h}(\vec{x}) \end{aligned} \tag{3.95}$$

with the state vector $\vec{x} \in \mathcal{X}^n$, $\vec{u}$ some system inputs and $\vec{z}$ a measurement or system output.

We denote the gradient with respect to the state variables of the zero order Lie derivative of h as $\nabla L^0 h$, and its first order derivative with respect to f as $\nabla L_f^1 h$. Higher order derivatives follow as described in (Hermann and Krener (1977)) and beforehand in this dissertation.

Let the corresponding observability matrix for Σ be $\mathcal{O}$ and span r dimensions. Then the system has $n - r$ unobservable modes. According to (Hermann and Krener (1977); Martinelli (2011)) we can define a dual system Σ' with r states in $\mathcal{R}^r \subseteq \mathcal{X}^n$. Then $\mathcal{O}'$ of Σ' has column full rank and all unobservable modes of Σ are orthogonal to Σ'. Conversely, all observable modes of Σ are linearly dependent (i.e. parallel) on Σ'. Fig. 3.11 illustrates a

2D example. The original system $\Sigma(X)$ is in the two-state space $X = [x_1\ x_2]$. We assume it has rank 1: $\text{rank}(\mathcal{O}) = r = 1$. Consequently, its nullspace is of dimension 1 representing the continuous symmetry w_1. The dual system $\Sigma'(X')$ only spans one dimension $X' = x'$ and its observability matrix $\mathcal{O}'$ has rank 1. Since $\Sigma'(X')$ is fully observable, its observable mode x' is orthogonal to the unobservable mode w_1 of $\Sigma(X)$.

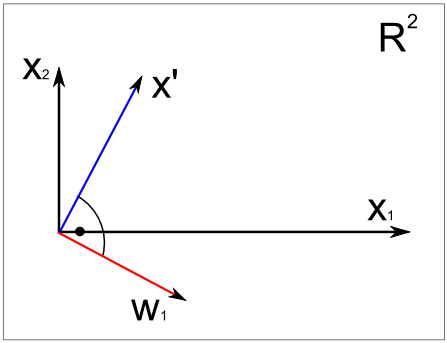


Figure 3.11: 2D example depicting the notion of an unobservable system $\Sigma[x_1, x_2]$ with one continuous symmetry w_1 and its dual system $\Sigma'[x']$ with its observable mode orthogonal to the unobservable mode of the original system Σ.

For a multi-sensor system we have for each sensor one (or possibly multiple) measurements $\vec{z}$. Without loss of generality we consider one measurement $z_m = h_m(\vec{x})$ per sensor m.

If we analyze the system Σ per sensor individually (i.e. we only mount one sensor at a time on the system) we have m non-linear observability matrices $\mathcal{O}_m$ to analyze. If one of them has coloumn-full rank, the system is is observable no matter what additional sensors we add.

On the other hand, if each sensor only provides partial observabilty all matrices $\mathcal{O}_m$ are rank deficient. If $r_m = \text{rank}(\mathcal{O}_m)$, each $\mathcal{O}_m$ spans a null space of dimension $z_m = n - r_m$ and has thus z_m symmetries orthogonal to each other as described in (Martinelli (2011)). On the other hand, each of the systems Σ_m with sensor m provides a set of observable modes $g^i_m(\vec{x}) \in \mathcal{G}_m$ with $\mathcal{G}_m \subset \mathcal{X}^n$ and $i = 1 \dots r_m$.

This means each sensor m provides a set of observable modes $\mathcal{G}_m$ that are *independent* of other sensors observability contribution. If the union $G = \bigcup_m \mathcal{G}_m$ of all subsets $\mathcal{G}_m$ has n orthogonal components, then G spans $\mathcal{X}_n$ which is a Hilbert space. Using Zorn's lemma and the Gram-Schmidt process one can show that every Hilbert space admits a basis and thus an orthonormal basis; furthermore, any two orthonormal bases of the same space have the

same cardinality. From this, and since every $g_m^i(\vec{x})$ in G is an observable mode *independent* of other sensors observability contribution, it follows that the orthonormal modes $\vec{x}$ are also observable.

We now have to show that G of all subsets $\mathcal{G}_m$ has n orthogonal components. For the observability matrix $\mathcal{O}_m$, $\text{ns}(\mathcal{O}_m)$ spans the nullspace and $\text{im}(\mathcal{O}_m)$ spans its image or observable space. The latter is also spanned by the set $\mathcal{G}_m$. Hence all $g_m^i(\vec{x}) \in \mathcal{G}_m$ for $i = 1 \dots r_m$ have orthogonal components to each other. This is not surprising since it is the observability analysis of the single sensor m. A second sensor m' makes the full state $\vec{x}$ observable if and only if $\text{im}(\mathcal{O}_{m'})$ has z_m orthogonal components with respect to $\text{im}(\mathcal{O}_m)$. This is fulfilled if

$$\text{rank}([\text{im}(\mathcal{O}_m), \text{im}(\mathcal{O}_m)]^T) = n \tag{3.96}$$

or, equivalently, if $\text{rank}(\mathcal{O}_m) + \text{rank}(\mathcal{O}_m) \geq n$ and

$$\text{rank}([\text{ns}(\mathcal{O}_m), \text{ns}(\mathcal{O}_m)]^T) \geq 2 * \min(\text{rank}(\text{ns}(\mathcal{O}_m)), \text{rank}(\text{ns}(\mathcal{O}_{m'}))) \tag{3.97}$$

with $\text{ns}(\mathcal{O}_m)$ being the nullspace of the observability matrix $\mathcal{O}_m$. The requirement expressed in the nullspace of the observability matrices arises from

$$\text{rank}(\mathcal{O}_m) = \text{rank}(\text{im}(\mathcal{O}_m)) + \text{rank}(\text{ns}(\mathcal{O}_m)) = n \tag{3.98}$$

and thus $\text{im}(\mathcal{O}_m) \perp \text{ns}(\mathcal{O}_m)$. Hence, the nullspace of the "less" observable system has to have the same number of orthogonal component with respect to the "more" observable system than the latter's rank is.

The above analysis shows which type of sensors yield which additional information to the system. This type of analysis is based on the suggestions of (Hermann and Krener (1977)) and operates thus in the time continuous and non-linear domain. Since measurements are thus assumed to be available at any given point in time, we are allowed to concatenate the observability matrices of each sensor.

Examples of Sensor Concatenations

We list the findings above in Fig. 3.12. We can now construct our real world sensors and directly obtain the observable states (see Example 1 below), or, vice-a-versa we can state our needs for the system and then evaluate which sensors we need (see Example 2 below). Furthermore, we can exactly predict

on which states a specific sensor acts and on which ones not at all. This is crucial for multi-sensor systems on which not all sensors are active at once. Situations may be an MAV using the camera as a pose sensor indoors and switching to GPS outdoors (see Example 3 below).

Example 1: We use the camera as a pose sensor consisting of three position- and attitude-drifting orthogonal 1D position sensors and 1 attitude-drifting attitude sensor. From table Fig. 3.12 (line 3) we see that the position sensors already render the whole system self-calibrating for their states and stable in roll, pitch and z velocity. The additional attitude sensor is responsible for the attitude inter-sensor calibration which it introduces (line 12 in Fig. 3.12). For the MAV state, no new observables are added by measuring this attitude.

Example 2: We need the MAV to maneuver globally consistent in all 6DoF. Considering our findings, we chose from Fig. 3.12 three orthogonal non-drifting 1D position sensors (line 4 in the figure and eliminating the position drift states p_v^w). Global consistent attitude is already granted with 2 such sensors since one sensor only needs one additional orthogonal component in the attitude. Eliminating the position drift state renders the global position observable. Finally, the third sensor leads to the observability of the velocity and position in its direction. This essentially describes the GPS and IMU as minimal and *self-calibrating* sensor suite for globally consistent position *and* attitude navigation (line 15 of Fig. 3.12).

Example 3: We want to navigate from indoor to outdoor and back switching each time from vision based navigation to GPS based navigation. We use the camera as a pose sensor. From table Fig. 3.12 (line 16) we see that, while navigating indoors, our sensors are calibrated and we are stable in roll, pitch and velocity in z. When switching to GPS mode we know that we can expect a change in yaw and in position, all other states are already observable and thus consistent with the upcoming GPS measurements. For proper initialization we can navigate vision based to only two way-points where we record GPS measurements. These measurements are sufficient for initial yaw and position correction. After this initialization, we can switch to GPS based navigation or use both sensors, camera and GPS at the same time. Since all states are observable in GPS mode and the inter-sensor calibration state between camera and IMU are modeled as constant, switching back to indoor (vision based navigation) is straight forward.

sensor type	sensor axis	dim [obs]/[tot]	state p_w^i	v_w^i	q_{iw}^i	b_w	b_a	λ	p_i^s	q_i^s	q_v^w	p_v^w	α	β	joint observability [joint]/[unobs](sym)
global pos w/ att drift	x, y or z	17 / 25	J1/2	J2	J2	ok	ok	ok	ok	X	J2	yz:X x:J1	X	X	5 / 13 (J1:1, J2:7)
	x & y	21 / 26	J1/2	J2	rp:ok y:J2	ok	ok	ok	ok	X	J2	z:X xy:J1	X	X	7 / 12 (J1:2, J2:3)
	x & y & z	23 / 27	J1/2	xy:J2 z:ok	rp:ok y:J2	ok	ok	ok	ok	X	J2	J1	X	X	8 / 12 (J1:3, J2:1)
global pos w/o att drift	x or y	15 / 17	yz:X x:J1	yz:X x:ok	J2	ok	ok	ok	ok	X	X	yz:X x:J1	X	X	3 / 5 (J1:1, J2:1)
	z	15 / 17	xy:X z:J1	xy:X z:ok	rp:ok y:U	ok	ok	ok	ok	X	X	xy:X z:J1	X	X	1 / 2 (J1:1) + 1 unobs
global vel w/ att drift	vx, vy or vz	16 / 21	X	J1	J1	ok	ok	ok	ok	X	J1	X	X	X	4 / 9 (J1:5)
	vx & vy	19 / 21	X	J1	rp:ok y:J1	ok	ok	ok	ok	X	J1	X	X	X	5 / 7 (J1:2)
	vx & vy & vz	20 / 21	X	xy:J z:ok	rp:ok y:J1	ok	ok	ok	ok	X	J1	X	X	X	5 / 6 (J1:1)
global vel w/o att drift	vx or vy	14 / 15	X	yz:X x:ok	J1	ok	ok	ok	ok	X	X	X	X	X	2 / 3 (J1:1)
	vz	14 / 15	X	xy:X z:ok	rp:ok y:U	ok	ok	ok	ok	X	X	X	X	X	1 unobs
global direction or attitude	dir w/ att drift	11 / 17	X	X	J1/2	ok	X	X	X	ok	J1	X	J1	J1	2 / 8 (J1:5, J2: 1)
	att w/ att drift	12 / 15	X	X	J1	ok	X	X	X	ok	J1	X	X	X	3 / 6 (J1:3)
	dir w/o at drift	10 / 13	X	X	J1/2	ok	X	X	X	ok	X	X	J1	J1	2 / 5 (J1:2, J2:1)
body vel	bvx, bvy or bvz	17 / 21	X	xy:J1 z:ok	rp:ok y:J1	ok	ok	ok	J3	J2	X	X	X	X	5 / 9 (J1:1, J2:1, J3:2)
specific sensors	GPS	19 / 19	ok	ok	ok	ok	ok	X	ok	X	X	X	X	X	---
	cam (pose)	27 / 31	J1/2	xy:J2 z:ok	rp:ok y:J2	ok	ok	ok	ok	ok	J2	J1	X	X	8 / 12 (J1:3, J2:1)
	cam (body vel)	20 / 21	X	xy:J1 z:ok	rp:ok y:J1	ok	ok	ok	ok	ok	X	X	X	X	2 / 3 (J1:1)

Figure 3.12: Sensor types with their observable states and system symmetries

3.4.7 Note on the Quality of Observability

We stated previously the three tracks of analyzing an EKF system like the one we propose in this chapter. First, the theoretical analysis of the time continuous and non-linear system, second, the theoretical analysis of the discretized and linearized system, and last the actual test on the implemented system. Whereas the latter is left for Chapter 4, the first has been widely discussed in this chapter. The second track mainly tackles questions on the quality of the observability and the hardware requirements to the sensors. Even though this goes beyond the scope of this dissertation we shed light to the possible methods. We propose a method to gain a first intuition on the quality of the observability. This method is still carried out on the continuous and non-linear system.

In order to have a measure of quality for the observability of a given state we define a state as "well" observable if, for the same input, the system output has a large gradient even if we only marginally change the state. Furthermore we make this requirement bijective: if the system output smoothly varies to a large extent, the gradient of state change leading to these variations is very small. That is, the measurements highly differ from each other even if the state only changed marginally. A state with such properties is very robust to measurement noise and is highly distinguishable within some proximity where this property holds. Hence, we call it "well" observable. Conversely, a state that leads to a small change in the output even though we smoothly change the state value to large extent is called in this dissertation as "poorly" observable. In the limit, the measurement does not change at all even if we move the state value from one limit to the other. Then, the state is called "unobservable".

Identifying the Quality of Observability

We follow the ideas published in (Hermann and Krener (1977)) and (Martinelli (2011)). For our analysis we consider the system Σ in (3.95) with the time continuous system inputs $u(t)$ and the time continuous system outputs (i.e. measurements) $h(x)$. A given input $u(t)$ on some time interval $[0, T]$ generates a state trajectory $x(t)$ and thus a certain output $h(x)$ in this time interval (input-output map). If a system is globally observable, then there are no two points $x_0(0)$, $x_1(0)$ in the state space S that have identical input-output maps for any control inputs. Locally weakly observability is given if, in a neighborhood of $x_0(0)$, there is no such point $x_1(0)$ with identical input-output map for a specific control input. We analyze our system on *locally weakly observability* and will denote in the following a locally weakly

observable system simply as *observable*.

The authors in (Hermann and Krener (1977)) state that, for an observable system, the state trajectory $x(t)$ ($t \in [0, T]$) and in particular $x(0)$ is uniquely defined by its inputs $u(t)$ and outputs $h(x)$. That is, all information needed to estimate the state of Σ in this time interval is contained in the propiosceptic (IMU for state propagation) and exteroceptic (measurements) sensors of our system. In turn, if we can reconstruct the state aid of this sensor information, the system is observable.

Without loss of generality, we use the following two assumptions in the remainder of the section:

- Assumption 1: The whole state space of the system Σ is observable. If we had a system Σ' with unobservable modes we can use the findings in (Martinelli (2011)) to identify these unobservable modes and design an observable system Σ in a reduced state space.

- Assumption 2: The trajectory of the robot is smooth (no kidnapping). The state evolution $x(t)$ and thus the measurements $h(x(t))$ can therefor be considered as smooth as well. Furthermore, because of assumption 1, we consider $x(t)$ and $h(x(t))$ as analytic.

The difficulty is to analyze $h(x(t))$ with respect to the influence of $x(t)$. First we write the system Σ in (3.95) in the control affine form:

$$\begin{aligned} \dot{x} &= \sum_i f_i(x(t))u_i(t) \\ z &= h(x(t)) \end{aligned} \tag{3.99}$$

with $u_0(t) = 1$. Then, we expand the output map $h(x(t))$ using the Taylor series (Röbenack (2004)):

$$h(x(t)) = h_0 + h_1 t + h_2 \frac{t^2}{2!} + \ldots + h_n \frac{t^n}{n!} \tag{3.100}$$

using the definition in (3.52) we can write the Taylor coefficients h_k as

$$
\begin{aligned}
h_0 &= h(x(t))\big|_{t=0} = L^0 h(x(0)) \\
h_1 &= \dot{h}(x(t))\big|_{t=0} = \sum_i h'(x(0)) f_i(x(0)) u_i(0) = \sum_i L^1_{f_i} h(x(0)) u_i(0) \\
h_2 &= \ddot{h}(x(t))\big|_{t=0} = \sum_{ij} L^2_{f_i f_j} h(x(0)) u_i(0) u_j(0) + \sum_i L^1_{f_i} h(x(0)) \dot{u}_i(0) \\
&\vdots
\end{aligned}
\tag{3.101}
$$

where we define $\dot{h} = \partial h / \partial t$ and $h' = \partial h / \partial x$. From (3.101) we see that higher order Lie derivatives only appear in the terms for higher order Taylor coefficients h_k. From the Taylor theorem, we know that h_0 contains most information of the output map $h(x(t))$ in the neighborhood of $x(0)$. More general, h_{i-1} contains more information than h_i of the output map $h(x(t))$ in the neighborhood of $x(0)$.

From (3.101) we can calculate all Lie derivatives using h_k and the derivatives of $u(t)|_{t=0}$. Also, the Lie derivative $L^i h$ does not affect $h_k \; \forall \; k < i$ and only contributes to $h_k \; \forall \; k \geq i$. More intuitively speaking, for a neighborhood $t \in (0, \epsilon)$ of $x(0)$ the higher order Taylor coefficients h_k contain less information of the output map $h(x(t))$. Thus, higher order Lie derivatives contribute less to the overall change of the output map than lower order Lie derivatives. This is a direct result from the Taylor theorem.

Note that this is a relative measure in the state space of the system. Thus, let us consider two different states s_1 and s_2. Assume that the dimension of s_1 is only spanned by the Taylor coefficients of grade k_1 and higher whereas the dimension of s_2 is only spanned by the coefficients of grade k_2 and higher with $k_1 < k_2$. That is, an infinitesimal change in s_1 changes the output $h(x(t))$ around $x(0)$ more than an infinitesimal change in s_2. In turn, large changes in the measurements affect s_2 more than s_1 and hence, s_1 is more robust to measurement noise and is called "more" observable than s_2 in this dissertation.

With the above mentioned we obtain a general intuition about the quality of observability of each observable dimension. Simply speaking, the higher the order of the Lie derivative in which the dimension first appears, the less this dimension is observable. While this is a general statement for well behaving systems, it does not declare anything about the observability of dimensions first occurring in the same order of Lie derivative. Furthermore, in the real system, we might have large noise on a well observable state and very low noise on a poorly observable state. In this case, the quality of the state estimate may be reversed with respect to our findings in this section.

Also, the quality of observability depends on the state x_0. Nevertheless, the above method shows which dimensions are generally more robust to noise.

3.4.8 Note on the Linearized and Discretized Systems

In this chapter we discussed the task of analyzing the time continuous non-linear system. As mentioned before, we divide a system analysis in three tracks. First, the analysis on the time continuous, non-linear system; second, the analysis on the linearized and discretized system; and finally the performance evaluation on the real system. We omit the second track in this dissertation and provide in Chapter 4 results in simulations instead. The performance on the real system is as well showed in Chapter 4.

In this section, we briefly highlight the following relevant questions to be answered in the second track:

- Quality of observability for each state
- Optimal trajectory for fast filter convergence
- Optimal sensor frequencies

For a given trajectory and given noise levels and frequencies per sensor the quality of observability can be defined by the certainty-gain per state. If we denote with P the state covariance matrix then $\vec{q}$ =diag$(P_k - P_0)$ shows the improvement on the certainty of each state after some time T. To define the most observable state relative to all states in the state space, $\vec{q}$ has to be normalized accordingly (Mourikis (June 2008)). To define the most observable dimension in the current filter step, we can observe the derivative of P: diag$(\frac{P_{k+1}-P_k}{t_{k+1}-t_k})$ and normalize it again accordingly.

Closely related to the quality of observability is the question on the optimal path for the system in control space for fastest filter convergence. In (Martinelli and Siegwart (2006)) the authors proposed a closed form solution for an optimal trajectory within a given time interval $[0, T]$ based on variational calculus. However, the solution to the corresponding Riccatti equation becomes impossible for large systems. Thus, we refer for this task to (Brian J Julian (September 2011)) and sketch briefly the different steps for a locally optimal movement. On each filter update (i.e. measurement $z = h(x)$) we can define the gained knowledge by analyzing the mutual information. In general, we want a measurement to minimize the conditional entropy $H(x|z)$ of the two random variables z and x:

$$H(x|z) = H(x) - I(x, z) \tag{3.102}$$

which is equivalent to maximize the mutual information between the two variables. Simply speaking, we want the measurement to provide as much information to the state as possible. Using this information together with the incrementally gained certainty in P per state leads to a metric if a state is sufficiently converged. We remember that $x = x(x(t), u(t))$ is actually dependent on the previous state and the input applied. As stated in (Brian J Julian (September 2011)) we can define a utility function

$$U := I(x, z) \tag{3.103}$$

Maximizing this utility function yields most information for fast filter convergence. According to (Brian J Julian (September 2011)) we can define the control law reflecting the gradient of this utiliy function

$$u_i = \gamma \frac{\partial U}{\partial u} \tag{3.104}$$

The control input u_i ensures a direction towards maximizing $I(x, z)$. The obtained inputs may contradict the vehicle dynamics and/or the task directives. Thus, a suitable control vector projection may have to be applied.

In real systems, the sensors applied have not infinitesimal fast measuring frequencies and it is a well known fact, that the discretization of a system may render it unstable. In (Mourikis (June 2008)) the authors propose a method to determine the optimal frequency within some constraints each sensor has to provide its measurement. The method is based on the discretization of the sensor noises and their consideration in the solution of the Riccatti equation of the system.

Again, the above only sketches the direction of possible ways to answer the relevant questions arising when analyzing a discretized and linearized system. The detailed approaches would go beyond the scope of this dissertation.

3.5 Conclusion

In this chapter, we have presented a novel loosely-coupled vision-inertial framework for simultaneous online calibration of all sensor parameters and extrinsic transformations, and long-term estimation of the system's egomotion in unstructured and potentially large-scale environments.

In particular, our approach loosens the assumption in state-of-the-art sensor-calibration approaches of having the vision frame rigidly aligned with the world reference frame. This novel approach of decoupling the frame in which the camera perceives the features from a world reference frame

allowed two major steps towards real-world power-on-and-go systems. As a first step, we can now treat the visual pose estimator as a black box yielding an unscaled 6DoF pose. This renders our approach independent of the type of visual pose estimator. More precisely, we are not bound anymore to the computationally expensive EKF based visual pose estimators. Instead, we can use the more appropriate and less computationally expensive key-frame based structure-from-motion or visual-odometry solutions. visual-odometry approaches can be implemented on platforms with very limited calculation power and are not bound to a finite working space. This makes our approach scalable to robot navigation in large environments even for platform with low calculation power. Moreover, we propose a contribution to address failure and drift estimate issues arising when treating the pose estimation part as a black box. We applied a non-linear observability analysis to show that even with an IMU and single camera only, we are able to navigate drift free in roll and pitch through large environments. This is particularly interesting for airborne vehicles.

A second step towards real-world power-on-and-go systems using our loosely-coupled approach is the fact, that we can now seamlessly add additional sensors to the core system in order to eliminate different drift states. The drifting terms are isolated by a concise observability analysis, and we provide extensive examples for further inclusion of additional drift-compensating sensors, such as a visual compass, a three-axis magnetometer, a sun sensor, or a GPS-receiver.

We used these specific sensors to introduce our general analysis of different sensor types used with an IMU as propagation sensor. With Fig. 3.12 we provide a table for quick sensor concatenations. The table provides information, first on the observable states and second on the unobservable states and their inter-connectivity with different continuous symmetries. The first shows that even with 1D sensors we are usually (except for 1D body-velocity sensors) able to estimate the introduced inter-sensor calibration states, the IMU biases and possible scale factors. The second information gives exact knowledge on which type of sensors we need to add to the system to render specific states observable. It also provides deep insight to which sensor affects with (MAV) states. This is crucial for sensor switching strategies in flight (for example switching from vision only navigation to GPS navigation) since we can then perform an adapted initialization to maintain the flight stability of the MAV.

We finally gave a short insight on the quality of the observability by analyzing the non-linear time continuous system and saw that the observable dimensions covered by lower order Lie derivatives are (generally speaking)

better observable. We define a state to be well observable if the measurements (i.e. system outputs) experience a large change, even though the state only changed marginally. Thus, the state is more robust to noise.

All together, this chapter suggests a modular framework for long-term navigation on power-on-and-go systems. Using a loosely-coupled approach, the system uses very low calculation power while still self-calibrating all sensor parameters and extrinsic transformations and estimating the robot's pose in a statistical optimal way.

Chapter 4

Results: Performance Evaluation

In the previous chapter we provided the theory for our modular multi-sensor approach for on-board micro helicopter navigation in large environments. We identified the issues of the drifts which the different sensors have and showed an overview which sensors can be used to obtain a system which is drift free in the desired dimensions. Furthermore, we showed that one can generally omit calibration procedures. This property of self-calibration makes the system not only suitable for long-term tasks but also renders it into a power-on-and-go system without the need of previous calibration procedures.

While the theory discussed in the previous chapter is necessary for the system's observability analysis it is not a sufficient criteria for a discretized and linearized implementation. We implement our appoach in an Extendend Kalman Filter (EKF) framewok. Thus, we have to ensure that the approach still yields good performance after this discretization and linearization. As we will see in the simulation results, the non-linearities introduced by the camera (i.e. scale factor and attitude drift states) are considerably more significant than the non-linearities caused by the addition of drift compensating sensors (magnetometer and GPS receiver in our implementation). Thus, the behavior of the core sensor setup is similar to the two sensor setups augmented by a magnetometer and/or a GPS receiver discussed in Section 3.4. All of these three setups are considered in the qualitative evaluation in a large scale outdoor environment.

We divide this chapter into the following three sections. First, in Section 4.1, we test our implementation on simulated data. We evaluate the behavior on state initialization values off the true value and identify a sig-

nificant sensitivity of the scale factor to wrong initializations. We also shed light to the sensitivity of different amount of excitations. We analyze the three systems: the core-sensor suite (camera and IMU), the core-sensor suite augmented by a magnetometer and the core-sensor suite augmented by a GPS receiver. Second, in Section 4.2, we quantitatively test our core sensor suite on-board a real micro helicopter (FireFly from AscendingTechnologies [1]. To this end, we implemented the EKF framework in a distributed way on an ARM7 and ATOM 1.6 GHz processor board. This enables us to control the MAV in position, velocity and attitude at 1 kHz. Finally, in Section 4.3, we show the performance of our our approach in a real-world scenario in a large outdoor environment and demonstrate flights in an altitude range from 0 m (i.e. vision based landing) up to 70 m above ground. The distances flown purely vision based reach up to about 360 m. This is a qualitative test because of the lack of ground truth data – we compare the filter estimates to the raw GPS data. In these tests, we also show the switching from one sensor suite to another *while* the MAV is flying.

4.1 Simulation Results

Even though we proved observability of the three base systems discussed in Section 3.4 it is crucial that EKF frameworks are tested in simulation and on real data. Even in simulation the idealized zero mean, white Gaussian noise is deformed in the non-linear equations. Furthermore, an EKF linearizes around the current working point. Thus, it is necessary to show that this linearization holds. Then, the experiments on real data show the robustness against modeling errors.

In this section, we evaluate the filter in simulations with respect to different initialization errors and system-input excitation-levels. For each simulation run, we change the initial value of one parameter with respect to its true initial value and keep the others with correct initialization. More precisely we analyze the distance between IMU and camera p_i^c, their rotation q_i^c, the scale factor λ, and the acceleration and rotational rate biases of the IMU b_a and b_w respectively. We also alter the initialization values for the rotational offset between the visual frame and the world frame q_v^w. For the systems described in Section 3.4.2 and 3.4.3 we additionally change q_i^m, α and p_v^w, p_i^g respectively. The true initialization values are arbitrarily chosen and are listed in table 4.1. A simulation run lasted 100 simulated seconds with accelerations of max $3\frac{m}{s^2}$ and a mean RMS value of $1\frac{m}{s^2}$ with respect to zero acceleration.

[1] www.asctec.de

The rotational velocities are max $\frac{\pi}{6}\frac{rad}{s}$. For each state we falsify its initial value, we plot the RMSE values of all filter states. This gives insight on the convergence time of the different states. Also, such a plot shows if a wrong initialization of one state influences the convergence behavior of another. We plot a representative graph for all three systems together (core-sensor suite, augmented with magnetometer and augmented with GPS) unless a notable difference occurs. For the analysis on different excitation levels in acceleration, we used data-sets with an RMS value of $1\frac{m}{s^2}$, $2\frac{m}{s^2}$, and $3\frac{m}{s^2}$ each with an RMS in angular velocity of $0.5\frac{rad}{s}$. We also changed the excitation level in angular velocity to $0.2\frac{rad}{s}$ and $0.8\frac{rad}{s}$ (RMS) while keeping the acceleration excitation at $1\frac{m}{s^2}$ (RMS).

Table 4.1: Default simulation values

$p_i^c[m]$	$[0.1\ 0.5\ -0.04]^T$
$q_i^c[rad]$	$rpy[0.2\ -0.3\ 0.4]^T$
λ	0.5
$b_a[\frac{m}{s^2}]$	$[-0.1\ -0.2\ 0.15]^T$
$b_w[\frac{rad}{s}]$	$[0.01\ 0.02\ -0.015]^T$
$q_i^m[rad]$	$rpy[0.5\ -0.9\ 0.5]^T$
$\alpha[rad]$	0.7255
$p_v^w[m]$	$[1.1\ -2.1\ 3.1]^T$
$p_i^g[m]$	$[0.5\ -1.1\ 1.5]^T$

The simulation data is based on properties of real-world sensors. We assumed a visual pose estimate at 20 Hz and a Crossbow VG400CC-200 IMU providing data at a rate of 75 Hz. The noise contained in the acceleration measurements amounts to $0.5\ \frac{m}{s\sqrt{h}}$ and the one of the gyroscopes to $4.5\frac{deg}{\sqrt{h}}$. We assume to have accurate timestamps from the sensors to deal with time delays.

The faster sampling rate of the IMU compared to the vision system demands for a multi-rate handling in the practical implementation. We propagate the filter state and its error covariance each time an IMU measurement is obtained as described in Section 3.1.1. We perform an update step upon a vision reading. Our simulated data reflects this multi-rate aspect as well as the noise properties of the real hardware.

4.1.1 Simulations on the Camera-IMU Translation

For the distance between IMU and camera p_i^c we added 0.01m to 0.9m to the true initial value of each axis. Fig. 4.1 shows the RMSE of the states of the filter compared to ground truth. The higher the RMSE the longer the filter takes to converge. Sudden jumps to high RMSE values would reflect the destabilization of the filter because the initialization value was too far from the truth. Fig. 4.1 represents the behavior of all three systems discussed in Section 3.4. The plot also shows that p_w^i, v_w^i and b_a are affected by a wrong initialization of p_i^c. Fig. 4.2 shows the convergence behavior of these states. In the figure, we also see that other states, like the gyroscope biases b_ω are much less affected (note the different scales in the plots). Fig. 4.3 shows the influence on different excitation levels. We clearly see that higher excitation in angular velocities is favorable for fast state convergence. We did not notice notable differences in changing the acceleration excitation level.

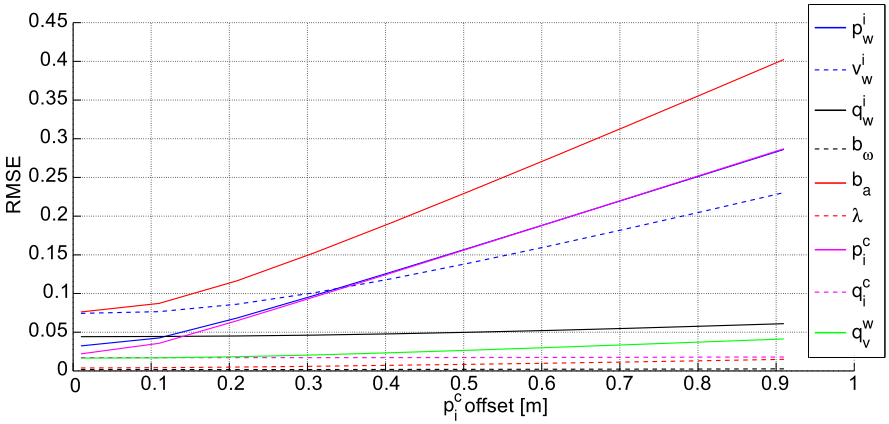


Figure 4.1: The graph shows the evolution of the state RMSE when the initialization error on p_i^c increases. A higher RMSE value indicates longer convergence time of the given state. Sudden jumps to very high RMSE values would indicate the destabilization of the filter because of the wrong initialization. Here we see that the wrong initialization of p_i^c also influences the convergence time of p_w^i, v_w^i and b_a.

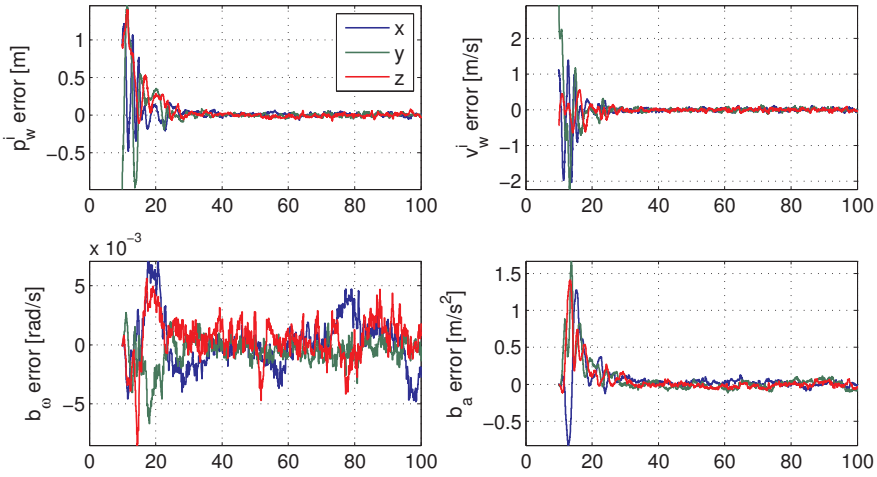


Figure 4.2: Wrong initialization of p_i^c also influences the convergence time of other states. In particualar p_w^i, v_w^i and b_a. Also, there are states such as b_ω which are barely influenced by a wrong initialization of p_i^c. Note the different scales in the plots

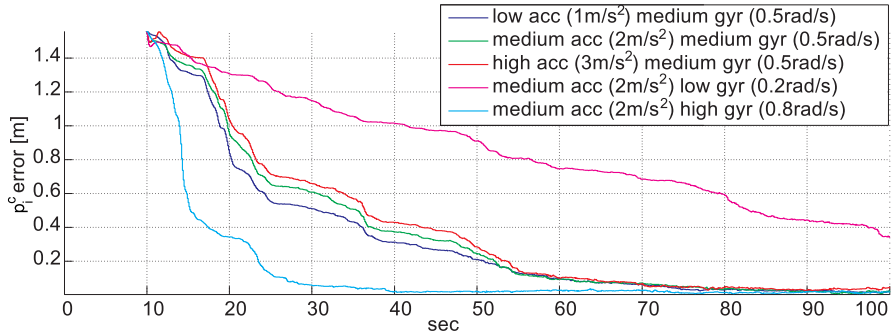


Figure 4.3: Different excitation levels on the system input (accelerations and angular velocities) influence the convergence behavior of the state p_i^c. While the influence on different angular velocity excitation is clearly visible, we did not observe any notable differences when changeing the acceleration excitation level.

4.1.2 Simulations on the Camera-IMU Rotation

For the rotation between IMU and camera q_i^c we added 0.01 rad to 0.9 rad to the true initial value of each axis. We summarize the performance in Fig. 4.4 for all three discussed systems. The plot shows that basically the convergence time of all states is affected except the one of b_ω and λ. Fig. 4.5 shows the state convergence behavior when different excitation levels are applied. We see no notable differences in the convergence behavior on different input excitation.

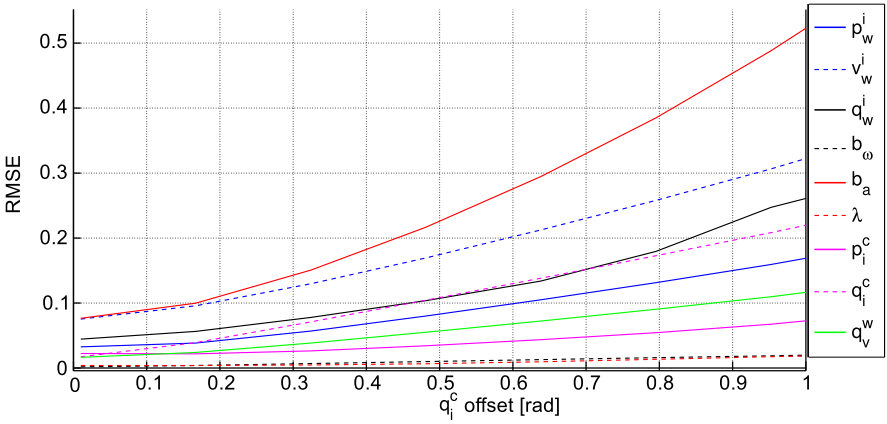


Figure 4.4: The graph shows the evolution of the state RMSE when the initialization error on q_i^c increases. A higher RMSE value indicates longer convergence time of the given state. Here we see that the wrong initialization of q_i^c also influences the convergence time of most other states except b_ω and λ.

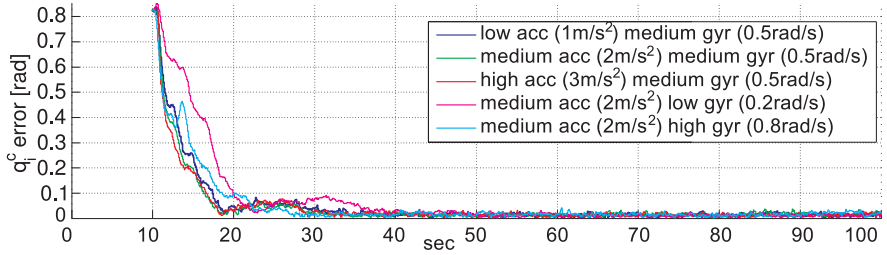


Figure 4.5: Different excitation levels on the system input (accelerations and angular velocities) influence the convergence behavior of the state q_i^c. While a different convergence behavior is barely observable, we can assume a higher excitation in angular velocities to be favorable for state convergence.

4.1.3 Simulations on the Visual Scale

We also altered the initial value of the scale from 0.1 to 5. The correct initial value is 0.5. Note that a good initial estimate of the scale is crucial to not destabilize the filter. In fact, we consider it as the only parameter which is crucial to be initialized correctly but is not easily measurable from the start on. Hence, in (Kneip et al. (2011)) we developed a method to robustly determine the initial scale and its covariance only based on three consecutive camera frames. Fig. 4.6 depicts the RMSE evolution for a wrongly initialized scale state. At the correct initialization value we observe an RMSE of λ below 0.004. We see that the RMSE is rapidly increasing with larger initialization errors. The behavior is similar on all three systems discussed in Chapter 3.4. All metric states p^i_w, v^i_w, b_a and p^c_i are influenced by a wrong scale initialization.

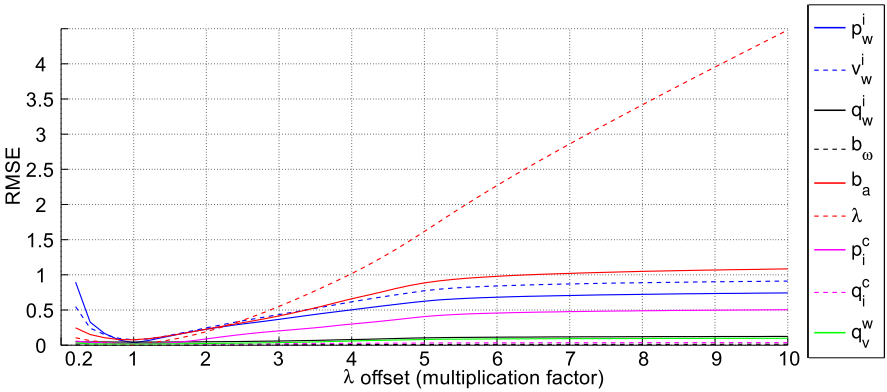


Figure 4.6: The graph shows the evolution of the state RMSE when the initialization error on λ increases. A higher RMSE value indicates longer convergence time of the given state. Here we see that a correct initialization of the scale is crucial. Naturally, all states containing metric information (p^i_w, v^i_w, b_a and p^c_i) are heavily influenced by a wrong scale estimate. The plateau in the right part of the plot indicates that the filter does not converge at all anymore. For this reason we developed in (Kneip et al. (2011)) a method to robustly estimate the initial scale with its variance by only observing three consecutive camera frames.

For the particular case of a wrong scale initialization we plot the temporal responses of the affected states in Fig. 4.7. We note that these affected states were initialized correctly at the beginning, nevertheless, they need some convergence time because of the wrongly initialized scale. Additionally, since Fig. 4.7 depicts representative behavior for all simulations performed,

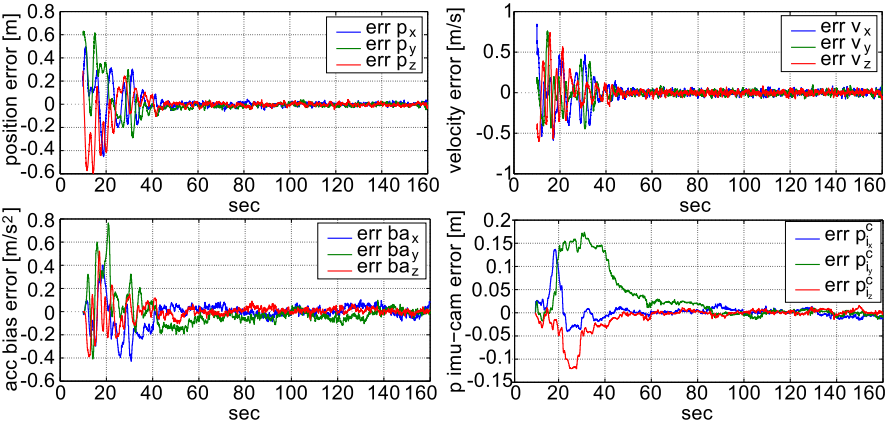


Figure 4.7: The graph shows the temporal evolution of the states affected by a wrong scale initialization (p_w^i, v_w^i, b_a and p_i^c). Note that all plotted states in this figure were initialized correctly but need some convergence time due to the wrong scale initialization. The temporal evolution also shows that the RMSE value in Fig. 4.6 represents good convergence with large initial value (in contrast to low initial value and large final value or bad convergence behavior).

it shows that the filter has a good convergence behavior for a reasonable initialization error.

Since the scale factor λ seems to be a delicate state with the need of correct initialization and prone to not converge for large initialization errors, we analyze also the effect of different excitations. In Fig. 4.8, we show the temporal evolution of the scale state with different levels of excitations. For each simulation, the scale was initialized 200% off its true value (i.e. at 1.5 instead of 0.5).

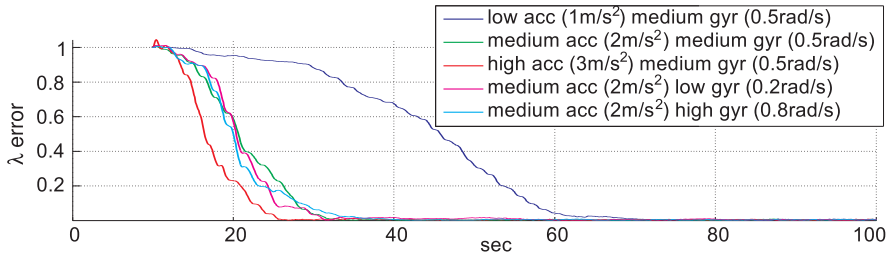


Figure 4.8: Different excitation levels on the system input (accelerations and angular velocities) influence the convergence behavior of the state λ. We see that a higher excitation in accelerations is favorable for state convergence.

Even though the scale state seems to be a delicate state for initialization, our framework is able to track it accurately on changes. This is key for long term robot navigation in large environments. Fig. 4.9 shows the temporal evolution of the scale state. We initialized the scale 50% off its true initial value (i.e. at 0.25 instead of 0.5) and let the true value rise linearly 100% of its initial value during the simulation time of 100 seconds. The RMSE value of 0.017 during the last 10 seconds of the simulation is in this case higher than in the above case with constant scale (RMSE for constant scale was below 0.004). In the plot, we notice the slight delay the estimated scale has with respect to the true scale value. This is due to our zero motion model of the scale change. We highlight that the scale state only has spatial drift. That is, while hovering on spot and observing the same features, the scale does not drift. The simulated scenario in Fig. 4.9 represents thus a constant movement always observing new features. The assumed 100% scale drift in 100 seconds (i.e. in 50m when moving at $0.5\frac{m}{s}$) represent a large drift in state-of-the-art algorithms. Of course, the amount the scale factor changes depends on the quality of the visual pose estimator (visual SLAM, visual odometry, etc).

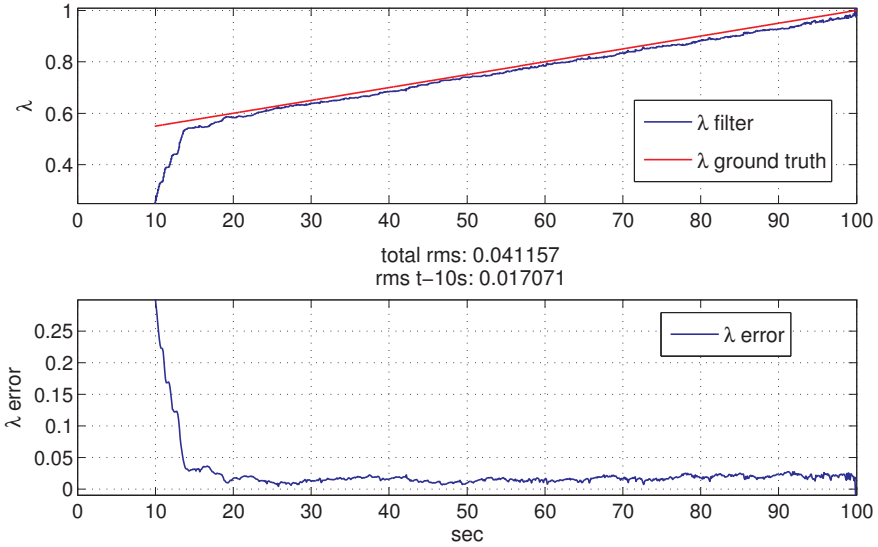


Figure 4.9: The graph shows the temporal evolution of the scale state λ. We initialized the scale at 0.25 instead of its true initial value of 0.5 and let the true value rise linearly more than 10% during the simulation time of 100 seconds. The RMSE value in this case is 0.017 in the converged phase (i.e. measured during the last 10 seconds). This is only slightly higher than in the case where the scale had a constant value (RMSE 0.004 in that case). Since we apply a zero motion model for the scale state, we notice a slight delay of the estimated scale factor with respect to the true scale factor. In practice, the scale factor only changes if new or different features are observed. That is it has a spatial drift and the simulated situation represents a movement of the robot always observing different features.

4.1.4 Simulations on the IMU biases

We analyzed the different influences of a wrong initialization in the IMU biases. Fig. 4.10 plots the RMSE curves. We see that a wrong initialization on b_a barely influences the other states. In contrast, wrong initialization on the gyroscope biases b_ω is critical for the filter stability. Initial errors greater than $0.2\frac{rad}{sec}$ destabilizes the filter. Fortunately, a good gyro bias estimate can be easily measured in all three axes at system start-up when the robot stands still. For the acceleration bias b_a estimation, we see in Fig. 4.11 that higher excitations in angular velocities are favorable for state convergence. Conversely, for the gyroscope bias estimation, no notable differences can be observed.

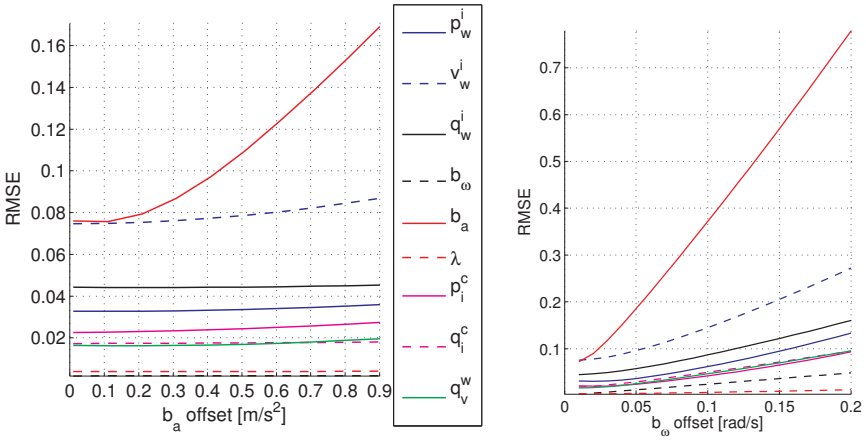


Figure 4.10: The graph in a) shows the evolution of the state RMSE when the initialization error on b_a increases. The graph in b) shows the evolution for a wrong b_ω initialization. A higher RMSE value indicates longer convergence time of the given state. Here we see that the wrong initialization of b_a barely influences the other states. However, all three filter designs discussed in Section 3.4 seem to be sensitive to a wrong b_ω initialization. We stopped the simulations at an initial error of $0.2\frac{rad}{sec}$ because the filter got unstable at higher initialization errors.

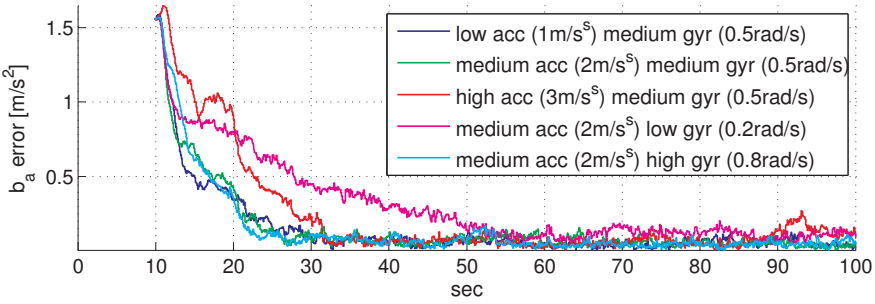


Figure 4.11: Different excitation levels on the system input (accelerations and angular velocities) influence the convergence behavior of the state b_a. A high excitation level in angular velocities is favorable for faster state convergence.

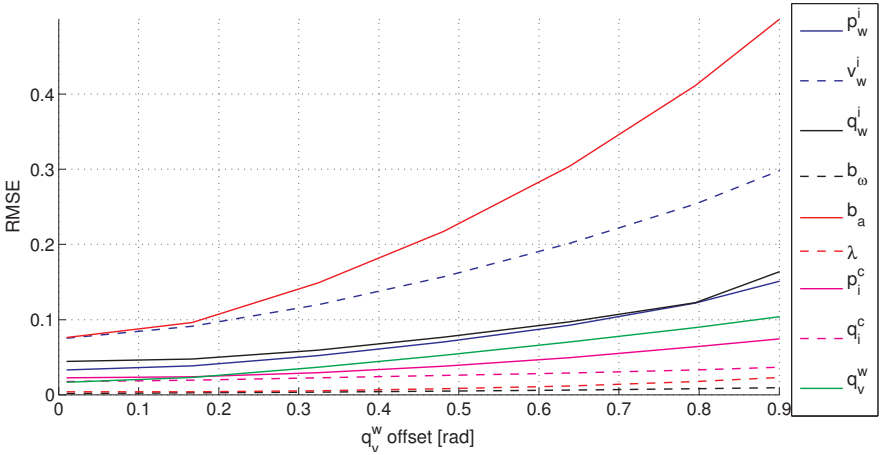


Figure 4.12: The graph shows the evolution of the state RMSE when the initialization error on q_v^w increases. A higher RMSE value indicates longer convergence time of the given state. Here we see that the wrong initialization of p_v^w also influences the convergence time of p_w^i, v_w^i and b_a.

4.1.5 Simulations on the Visual Attitude Drift

We added an initial error to q_v^w in the same way as we did to q_i^c. That is, we added an offset of 0.01rad to 0.9rad. Fig. 4.12 indicates that mainly the position, velocity and acceleration bias states are sensitive to an initial error on q_v^w. This is understandable since a wrong attitude between the vision and world frame directly affects the position estimate and its derivatives.

In order to ensure long term navigation in large environments we tested the filter framework on the capability to track a drift in the attitude between the vision and world frame. To this end, we applied a drift of more than 10% to the roll and pitch drift between the two frames. Fig. 4.13 depicts the temporal evolution of the attitude drift state q_v^w. We initialized the initial value slightly off the true value. After a few seconds the estimated value converged to the true value and followed the changes accurately. This capability is key in order to not crash an aerial vehicle. Airborne robots are particularly sensitive to wrong roll and pitch estimates. If the visual pose estimator drifts in these values, our approach proves to be capable to identify this drift and to compensate accordingly.

4.1.6 Simulations on the Additional States

For the additional states p_i^m and α introduced in the system described in Section 3.4.2 we did not see any notable issues when changing the initial values. We noticed the positive influence on the state convergence time of α on higher excitations in angular velocities (Fig. 4.14. For the GPS measurements we applied a large measurement noise (standard deviation of 0.5m) to account for the inaccurate reading in real world. The states p_i^g and p_v^w had a long convergence time, however, even large initialization errors did not destabilize the filter. The fact that the additionally introduced states p_i^m, α, p_i^g and p_v^w do not quickly destabilize the filter lies in their function to only account for corrections on slowly and spatially drifting states of the core-sensor suite.

Note on Most Observable Modes

In all RMSE analysis plots in this section we note the fact that the velocity of the IMU v_w^i and the bias of the accelerometers b_a show the largest error and are most sensitive to any wrong initialization of other states in the state space. Going back to Section 3.4.7 we see that all states except v_w^i, b_a, and b_ω are part of the subspace spanned by the zero order Lie derivatives $\nabla L^0 h_1$ and $\nabla L^0 h_2$. Whereas v_w^i and b_ω are part of the subspace spanned by the first order Lie derivatives and b_a are part of the subspace spanned by the

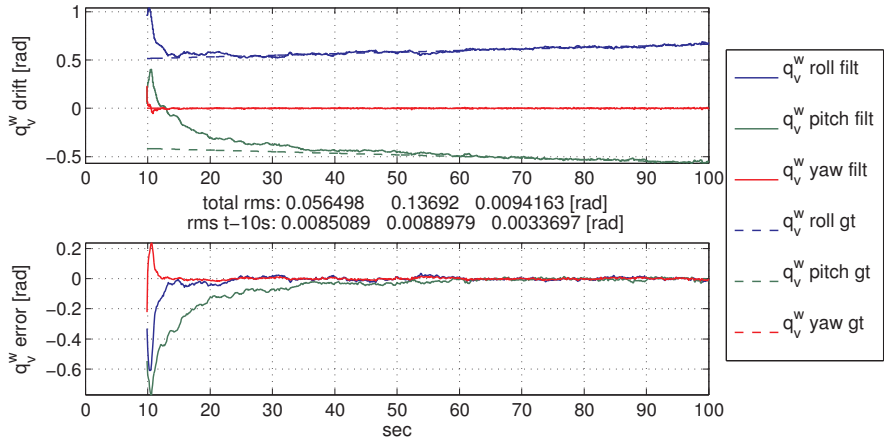


Figure 4.13: The graph shows the evolution of the attitude drift state between the world and vision frame. Even with slightly wrong initialization, our approach is capable to estimate the correct drift values and track them accurately. In this simulation, we applied a 10% linear drift on roll and pitch between the world and vision frame during the simulation period of 100 seconds.

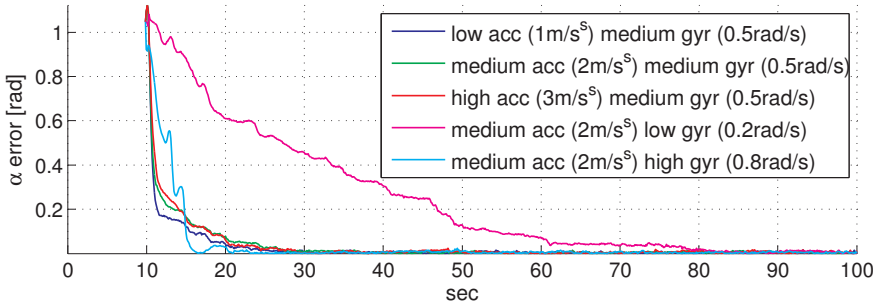


Figure 4.14: Different excitation levels on the system input (accelerations and angular velocities) influence the state convergence behavior. Influences on the convergence behavior of α on different excitation levels are barely visible. However, a higher excitation in angular velocities is favorable for faster state convergence.

second order Lie derivative. According to our findings in Section 3.4.7, for a general motion, b_a should be less observable than v_w^i and b_ω and than the remaining states. We recall here that we refer to a state to be *less* observable than another, if this state generally contributes less to $\nabla_{x(0)} h(x(t))$ than another state when moving an infinitesimal step in the state space. The RMSE values of b_a with respect to v_w^i and both with respect to the remaining states reflect this hierarchy of quality of observability also in our simulations. The low RMSE value of b_ω is due to the low noise applied to the gyroscope input (about factor 80 lower than on the accelerometers). This mirrors the situation we discussed in Section 3.4.7: a less observable state with very low noise may yield better accuracy than a well observable state with large noise.

4.2 Delay Compensated Distributed Implementation

In this section, we present a versatile implementation of our modular sensor fusion approach described in Chapter 3 to enable autonomous flights of a Micro Aerial Vehicle (MAV) which has only slow, noisy, delayed and possibly arbitrarily scaled measurements available. Using such measurements directly for control would be practically impossible as MAVs exhibit great agility in motion. This section and the work within is published in (Weiss et al.

(2012a)).

We demonstrate that the proposed framework is capable of running entirely on-board a MAV boosting its autonom while performing state prediction at the rate of 1 kHz. The results illustrate that our delay-compensated approach is able to handle measurement delays (up to 500ms), noise (std. deviation up to 20 cm) and slow update rates (as low as 1 Hz) while dynamic maneuvers are still possible. We present a detailed quantitative performance evaluation of the real system under the influence of different disturbance parameters.

The assessment of the performance and practicality of this EKF framework is crucial as it can only be performed following its actual implementation on a real system – the true effect of all the assumptions revealed and any inaccuracies due to modeling/linearization are identified only then. The most relevant caveats are the assumptions of a Gaussian, linear and discrete world. Even though the sensor readings may be well-approximated by Gaussian distributions, they certainly get distorted after applying the non-linear equations. Moreover, discretization effects may lead to instability of the system. Hence, we conduct a thorough quantitative evaluation to reveal consistent operation of a highly robust and modular filtering framework, running online and on-board a MAV.

The aim of this section is to provide a versatile and modular EKF framework which tackles pose estimation *and* inter-sensor calibration not only as a theoretical construct, but also realize this on a working MAV system operating in real-time. Our approach achieves state prediction at 1 kHz while the framework is kept modular to allow any kind of sensor measurements to be incorporated quickly (Section 3.4).

4.2.1 System Setup

The MAV we use is a prototype of a hexacopter from Ascending Technologies[2] (AscTec). It is a helicopter driven by six rotors, symmetric to the center of mass. The control is performed solely by changing the rotational velocity of the rotors. The key features are up to 500 g payload, improved vibration damping for the sensors and the Flight Control Unit (FCU) "AscTec Autopilot". This FCU features a complete IMU and two 32 Bit, 60 MHz ARM-7 micro-controllers used for data fusion and flight control.

The so-called Low Level Processor (LLP) is considered as a black box which manages the hardware and performs IMU-sensor data-fusion for attitude control. The other micro-controller, the High Level Processor (HLP) is

[2] www.asctec.de

dedicated for custom code. IMU data is provided at an update rate of 1 kHz via a high speed serial interface from the LLP. In particular, this comprises body accelerations, body angular velocities, magnetic compass, height measured by an air pressure sensor and the estimated attitude of the vehicle. On the HLP, a position controller based on nonlinear dynamic inversion and reference model based set-point following is implemented which we described in detail in (Achtelik et al. (2011b)).

For the computationally more expensive on-board processing tasks, we outfitted the helicopter with an on-board 1.6 GHz Intel Atom Based embedded computer, available from Ascending Technologies. This computer is equipped with 1 GB RAM, a MicroSD card slot for the operating system, 802.11n WiFi, a Compact Flash slot, USB 2.0 interfaces for external sensors and serial ports for communication to the HLP. We run a standard Ubuntu Linux 10.04 on our on-board computer and use the ROS [3] framework as a middle-ware for communication, parameters and monitoring of our processes.

Also crucial for the whole system to work is accurate time synchronization between all involved parts. Therefor, Atom-computer and HLP time synchronization packets are exchanged. Transfer delay and jitter are then estimated in a Network Time Protocol (NTP) like way.

4.2.2 Distribution of Processing Tasks

The computational burden of our EKF sensor-fusion framework presented in Chapter 3 is in the prediction of the covariance matrix. Note that the measurements only imply a matrix inversion of the size of the number of measurements (e.g. 6×6 for a 6DoF pose sensor or 3×3 for a 3DoF body-velocity sensor). In addition, the update frame-rate is usually much lower than the prediction frame-rate[4]. In our approach, we define three processing tasks: (a) the pure state prediction as in (3.20) to (3.24), (b) the covariance prediction as in (3.37) and (c) the full update step.

Task (a) is the least computationally demanding while it yields the most recent best guess of the system state (i.e. robot pose). For controlling a MAV, this is the most (time) critical part and should be executed at high rates with the least possible delay to allow fast response to disturbances or to enable dynamic flights. Since IMU data is available with almost no delay at a rate of 1 kHz at the user programmable "high level processor" (HLP) of our MAV (see Section 4.2.1), we implemented this part on the HLP executed

[3] www.ros.org

[4] In our setup using the Leica laser tracker, we run the prediction at IMU frame-rate of 1Khz and the update at about 7Hz.

at a rate of 1 kHz. Note that this time critical part is ensured to be executed in real-time which softens some real-time constraints for the remaining tasks being executed on a standard non-real-time operating system.

The computationally more expensive tasks (b) and (c) are implemented separately on the on-board Atom computer. The HLP sends its state predictions to the Atom computer over a highspeed serial link, while the Atom computer sends back state corrections every time a measurement update arrives. Crucial for this to work is accurate time synchronization between HLP and Atom computer which we implemented in a lightweight NTP like approach. The distribution and data-flow can also be seen in Fig. 4.15.

As it can be seen in Fig. 4.15, the current state predicted by the HLP is exchanged only at a rate of 100 Hz. This means that we predict the covariance at 100 Hz while the state is predicted at 1 kHz on the HLP. This is due to bandwidth limitations of the data link between HLP and Atom computer and of the relatively high computation cost for predicting a $N \times N$ covariance matrix. While this approximation might be problematic in theory, it did not cause any problems in practice since linearization errors in the period of 10 ms are negligible.

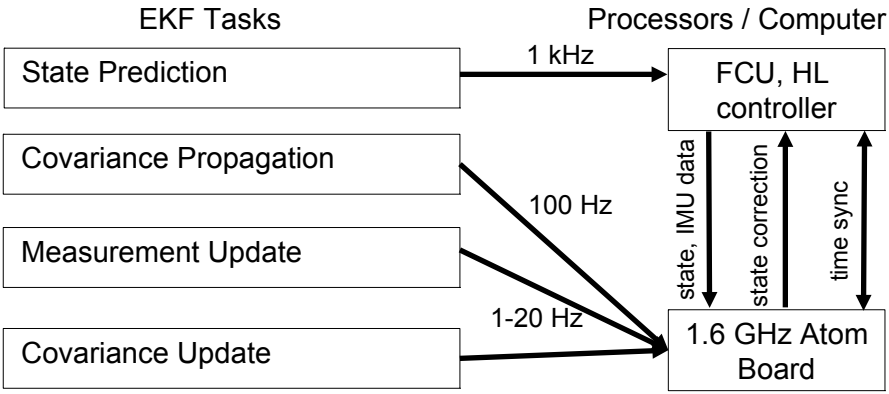


Figure 4.15: Distribution of the processing tasks with their execution rates. State prediction, as the most time-critical part for controlling the helicopter, is executed on the IMU micro-controller (HLP) and is guaranteed to run in real-time at 1 kHz. This leaves enough time for the more complex parts to be computed on the Atom computer with a non-real-time operating system. Also very important for the whole system to work is accurate time synchronization. (Weiss et al. (2012a))

4.2.3 Handling Measurement Delays

An efficient method to handle measurement delays is crucial for robust and accurate state estimation. Such delays usually occur due to computationally expensive processing tasks such as image processing on limited on-board hardware. However, the measurement update cannot be performed just after other processing is finished, instead it has to be aligned in time with the state prediction. One of the contributions of this implementation is a method that compensates for delayed measurements, ensuring both computational efficiency and theoretical exactness. Since both the HLP and Atom computer are synchronized, the timestamps of all sensors are referring to the same global time. Keeping a buffer of the S past states enables us to apply the obtained measurements at the exact time in the past they were taken. Thus the update is also theoretically exact, despite the delay. After performing an update step, the corrected state in the past is propagated again to the present time. The sequence in Fig. 4.16 depicts the process.

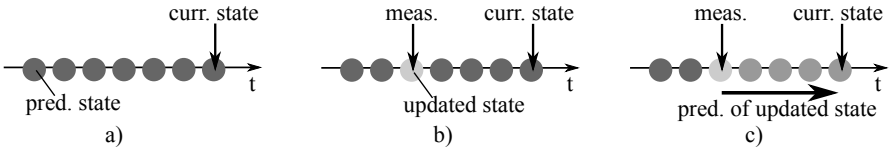


Figure 4.16: Method to handle a time-delayed measurement. a) a given situation in time, the robot controller uses the latest propagated state as reference. b) a delayed measurement arrives and corrects the corresponding buffered state in the past. c) from the corrected state on, we correct all states in the buffer until the most recent one by propagating the corrected state according to the system's propagation equations (3.20) to (3.24). (Weiss et al. (2012a))

Propagating the corrected state to the most recent state is computationally inexpensive for the state itself, but expensive for the covariance propagation. However, unless the most recent state uncertainty is used in other algorithms, there is no need to propagate the state covariance matrix to the present time. In fact, it is sufficient to have an up-to-date state covariance at the point in time where the latest measurement arrived. To minimize the computational effort and to avoid computation spikes, we suggest the following: when a measurement arrives, the state and covariance will be updated at the corresponding time of the measurement in the past. Then the state will be re-computed according to Fig. 4.16 until the present time. This re-computation is not expensive in terms of computing power.

During regular state prediction, each time we predict a new state, we propagate the covariance matrix in the past to the next state in the past.

That is, according to Fig. 4.17 while a state prediction e.g. x_6 is made, the propagation of the state covariance $P_{3|2}$ is computed.

If the measurements had a constant update rate and delay, the covariance prediction would be up-to-date at the correct point in the past where the latest measurement belongs to. This way, we distribute the covariance prediction optimally within the time between two measurements and have thus equally distributes processor load. In practice, the measurement delays vary slightly so we have to ensure an up-to-date covariance matrix by applying additional covariance prediction steps or ignoring redundant ones. This overhead is small in practical applications.

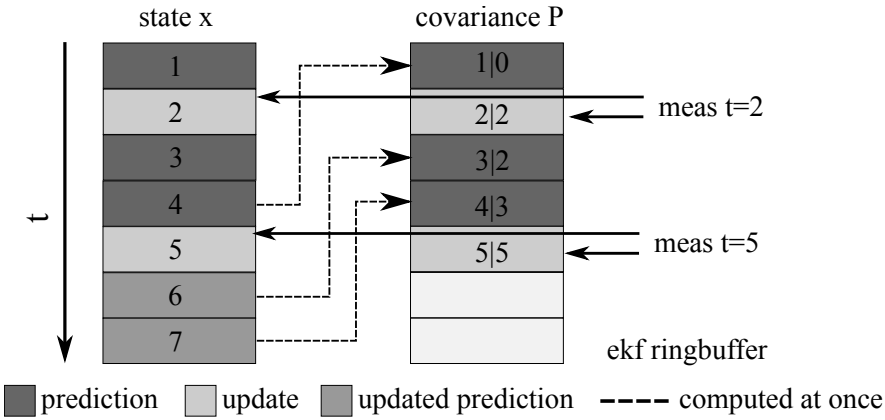


Figure 4.17: For the computationally efficient state covariance handling we use two ring-buffers: one for the states and one for the covariance. Each has its separate data pointer. At a certain time t the state buffer pointer represents the present time, whereas the covariance buffer pointer points at the time the measurement corresponds to. Both state and covariance are updated at that point of time and the state is propagated to its current data pointer (present time) according to Fig. 4.16. As further states get propagated, the covariance is propagated accordingly in the past (t=4 to t=7). Upon new measurement, covariance and state in the past get updated and again the state is propagated to its current data pointer. (Weiss et al. (2012a))

4.2.4 Handling Measurement Updates in Position Control

In the presence of low measurement update rates, noticeable correction steps occur, which in practice disturbs control of the MAV resulting in sudden excitations. Therefor, we still correct the state on the HLP in one step and perform state propagation from this correction, but we distribute the correction on the position controller over a constant period. In practice, smoothing the correction over 100 ms proved to be a good trade-off between precise set-point following and smooth motion of the MAV. The results at a low update rate of 1 Hz can be seen in the right part of Fig. 4.21: while the state estimate gets corrected immediately every second, the motion of the helicopter still stays smooth. For larger corrections, such as for loop closure events, correction smoothing would not be sufficient. Therefor, we suggest performing a correction without smoothing and at the *same* time set the set-point of the controller to the new corrected position. This avoids sudden movements and the large correction can be handled by a higher level task.

4.2.5 Modular Design

Our framework has a propagation part consisting of the propagation model using the dynamics acquired by the IMU and can be augmented with different types of measurement updates. This means, that this module remains unchanged using different sensors for measurement updates. This setup requires a modular software design capable of adapting to different and multiple sensors. Our ROS-based software implementation consists of a prediction core and allows different and multiple measurement modules to be added. Fig. 4.18 depicts the software architecture. In fact, to use another sensor to the system, it is only necessary to define the initialization values of the filter and the linearized measurement matrix H. The matrix H has to be defined such that the estimated measurement is $\hat{z} = H\vec{x}$ with $\vec{x}$ as the state vector. Furthermore, the residual $r = z - \hat{z}$ has to be calculated. This renders our approach versatile and reusable for a wide range of applications and different sensors.

4.2.6 Results on the Real Platform

This section presents our experimental evaluation of the proposed framework and implementation. We give a quantitative evaluation using a Vicon motion capturing system providing us ground truth. We also use the Vicon system to provide measurements for our experiments to cancel potential failures or

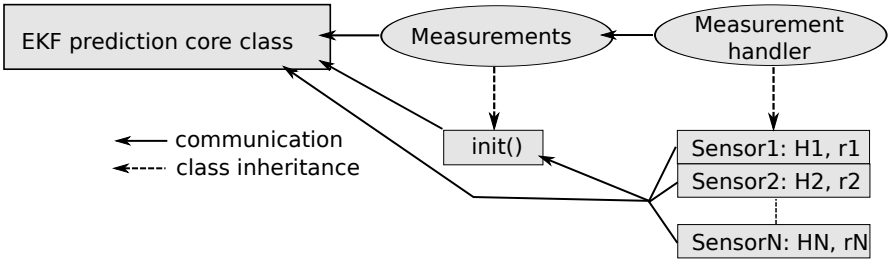


Figure 4.18: Software architecture of the proposed framework. The EKF prediction module of the framework remains unchanged when using different measurement sensors. We add a base class to handle measurements in general including an initialization routine. A dedicated measurement handler class represents then the different measurement sensors. Each sensor has its specific update – it is only necessary to set the initialization values, define the linearized measurement matrix H and calculate the residual $r = z - \hat{z}$ for each sensor added to the system. (Weiss et al. (2012a))

systematic errors of a pose-estimator such as (Visual-)SLAM. We reduce the rate of the Vicon measurements and add substantial noise and delay[5] in order to obtain realistic measurements. For each test, we change one of these parameters. The default values are measurements at 10Hz, Vicon standard noise (mm accuracy) and Vicon standard delay (negligible). The distance p_i^s from the IMU to the additional sensor is measured as $p_i^c = [0.015, -0.009, -0.028]$ and the corresponding attitude q_i^s as unit rotation. We conduct two main experiments: stationary hovering and a dynamic trajectory flight with all degrees of freedom of the helicopter involved. For all experiments, the state estimates (position, velocity and yaw) from the filter are used directly as input for the PID position controller (Achtelik et al. (2011b)) of the helicopter.

Stationary Hovering

First, we show the performance of the framework under hovering conditions. This is important to evaluate as the filter needs motion in order for all the states to be observable. Observability ensures no drift on these states. For this experiment, we let the vehicle hover at a height of 1 m for 30 s. We then compute the RMS error between the filter estimate and ground truth and between ground truth and the desired set-point. The latter is done in order to show the performance of the whole system including the position controller. In particular, we make the following evaluations:

[5]The tools we used can be found at http://www.ros.org/wiki/vicon_bridge

1. Ground truth against filter output for:
   (a) position p_w^i and orientation q_w^i in Roll Pitch Yaw (rpy) convention
   (b) scale λ of the position measurement in %
   (c) position p_i^s and orientation q_i^s in Roll Pitch Yaw (rpy) convention of the sensor[6] with respect to the IMU.
2. Ground truth against controller set-point: $x = 0$, $y = 0$, $z = 1$ m, yaw=0 rad. In the tables in the appendix, this is indicated by the second vector in the columns for p_w^i and q_w^i.

All these experiments are performed with a 6DoF pose sensor (e.g. camera) and a 3DoF position sensor (e.g. Leica Total Station). We denote these setups as with (w/) and without (w/o) attitude measurements respectively in the tables in the appendix.

Table A.1 in Appendix A.1 shows the results of distorting the signal with noise. We apply different noise levels on position and attitude ranging from a standard deviation of 5 mm and 0.5° to 20 cm and 2° respectively. It is clearly visible that position, attitude, scale and inter-sensor attitude are reliably estimated even under heavy noise conditions. Except for small noise levels, the estimate lies in the one-sigma bound of the measurement noise. For low noise levels the IMU noise dominates and thus the filter yields a larger bound. In the same table, it is also visible that the inter-sensor distance p_i^s is the least accurate estimate. The observability analysis requires this state to be non-zero. We assume that the small magnitude of our chosen p_i^s together with the low excitations in hovering may in practice not be sufficient for this state to converge fully. The results of low noise with- and large noise without attitude update can also be seen in Fig. 4.19.

In Table A.2 in Appendix A.1 we see the expected effect of delay introduced to the measurements. In our implementation, we have a state buffer of 2.5 s. In theory, adding a delay of 500 ms should thus have minimal influence on the filter performance. This is evident from the results in Table A.2. The slight RMS increase in position p_w^i and attitude q_w^i comes from the fact that for the current state, the last measurement update was up to 500 ms in the past. During this time, the filter state "integrated away" using the IMU prediction model only. Again, the state p_i^s seems to be barely observable in this configuration.

In Table A.3 in in Appendix A.1 we list the influence of the measurement rate on the filter performance. In theory, as long as we have information, the

[6] e.g. camera or laser scanner. In case of an external tracking system (Vicon, laser tracker), this is the pose of the reflector(s)

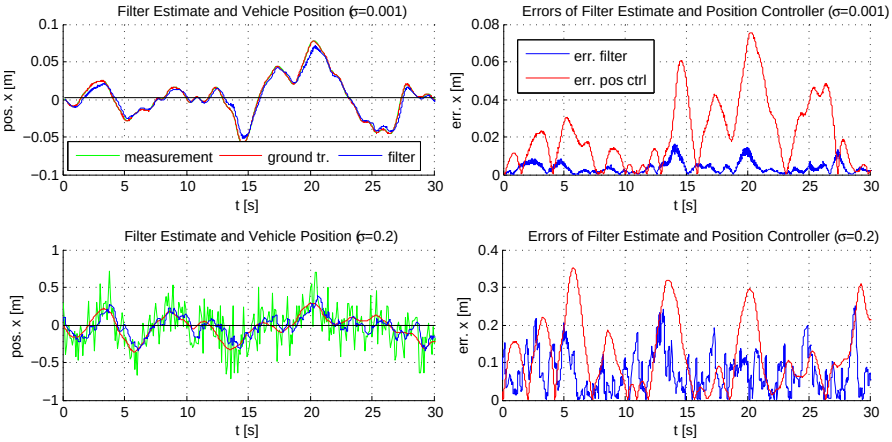


Figure 4.19: Performance comparison for the "hovering" experiment with (σ = 0.001 m) w/ attitude update (top) and ($\sigma = 0.2$ m) w/o attitude update (bottom). Left: ground truth position of the helicopter (red), the filter estimate (blue) and the measurement the filter was updated with. Right: error of the filter with respect to ground truth (blue) and error of the desired trajectory with respect to ground truth (red) and set-points. This is the overall error including the state estimator and the position controller of the helicopter. Note the different scaling of the plots. (Weiss et al. (2012a))

filter should converge. Again, the very slight increase of the RMS in position p_w^i and attitude q_w^i comes from the time between measurement updates where the filter can use the IMU prediction model only.

For all the experiments above, the scale estimate is very accurate. One may expect differently given that the scale is a hidden state and thus may compensate on the real system for linearization and non-Gaussian effects. We assume that the low excitations in hovering mode reduce these effects drastically while still giving enough information to actually observe the state.

Dynamic Flight

To show the performance of the filter and the whole system for dynamic flights, we let the helicopter fly a trajectory of the shape of a tilted ellipse (Fig. 4.20), keeping a track-speed of 1 m/s while performing a full 360° turn around the z-axis. We use this trajectory as input for the reference-model based set-point following. Based on the given position input, this model generates the accelerations and velocities (left part of Fig. 4.20) such that the helicopter is physically able to stay on the trajectory. The accelerations are used for feed-forward control whereas an error controller finally compensates for disturbances, keeping the helicopter on the desired trajectories for velocity and position.

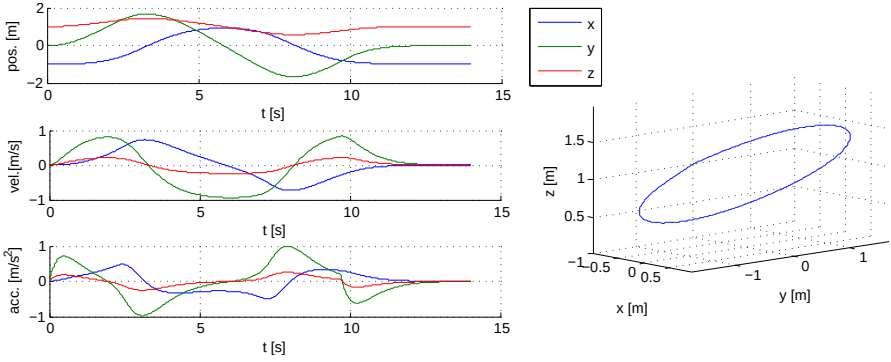


Figure 4.20: The trajectory used for the dynamic flight experiments. The top and right plot show the desired position. Middle and bottom plots show the velocities and necessary accelerations to keep an absolute track-speed of 1 m/s. (Weiss et al. (2012a))

Table A.4 in Appendix A.2 shows the results of different noise levels similar to the experiment in hovering mode. The values lie in the same range as for the hovering experiment if not slightly lower. Since the pose of the

filter is a direct measure, this result is expected. Moreover, the inter-sensor calibration states seem to have a lower RMS value as well compared to the hovering experiments. In dynamic flight, the requirements of having angular velocities and linear accelerations in at least two axes are fulfilled. As a rule of thumb, it is good to have as much excitation of the system as the Nyquist theorem is still fulfilled given the measurement sensor reading rate. Thus, Table A.5 and Table A.6 in Appendix A.2 yield the expected results. Naturally, because of the issue of "integrating away" of the states on large delays or low update rates, the RMS values of the position are higher in dynamic flight than while hovering. The last two tests in Table A.6 may reflect the above issue of not fulfilling Nyquist's theorem. Hence the RMS rises accordingly. The results of two dynamic flights with 10 Hz and only 1 Hz update rate can also be seen in Fig. 4.21.

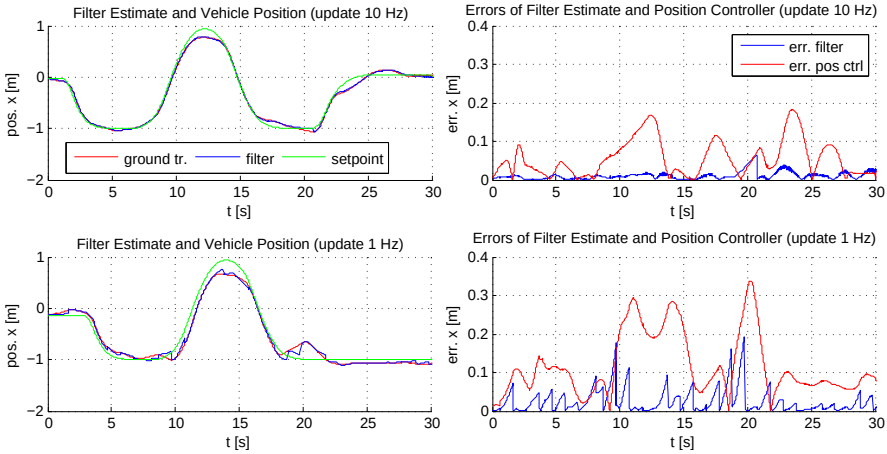


Figure 4.21: Performance comparison for the "ellipse" experiment with 10 Hz (top) and only 1 Hz (bottom) update rate. Left: desired trajectory (green), ground truth position of the helicopter (red) and the filter estimate (blue). Right: error of the filter with respect to ground truth (blue) and error of the desired trajectory with respect to ground truth (red) and set-points. This is the overall error including the state estimator and the position controller of the helicopter. (Weiss et al. (2012a))

4.2.7 Detecting and Handling Map Failure Modes

We discussed in Section 3.1.2 the possibility of detecting failure modes of the vision black box treated as 6DoF pose sensor. Furthermore, in Section 2.2, we presented a map-less approach based on inertial optical flow to use the camera as a scaled body-velocity sensor. We stated the motivation to use this approach to bridge map-losses, failure modes and re-initializations of the map-based approach (Section 2.1).

In this section, we use our inertial-optical flow body-velocity recovery approach discussed in Section 2.2 and merge this sensor type into our EKF framework discussed theoretically in Section 3.4.5. We now have both the approach to recover the scaled body-velocity and the theory how to fuse this information properly for adequate MAV control. More precisely, with this proper fusion we not only have an accurate gyroscope bias estimate for correct optical flow un-rotation (Fig. 3.10) but also a metric velocity information for MAV speed control and continuous self-calibration of the full sensor suite.

In the following, we show that the theoretical findings in Section 3.4.5 are also working in the implemented system. This shows the validity of this approach to initialize and bridge failures of a map based visual pose estimator. This subsection is published as a part of (Weiss et al. (2012b)).

Table 4.2 lists the values we used for the system in simulation, during which we ensured that the motion has excitation in at least two axes in acceleration and angular velocity (as advocated in Section 3.4.5). In practice, this is usually fulfilled due to the agile nature of the MAVs.

Table 4.2: Default simulation values

$p_i^c[m]$	$[0.1\ 0.5\ -0.04]^T$
$q_i^c[rad]$	$rpy[0.2\ -0.3\ 0.4]^T$
L	0.5
$b_a[\frac{m}{s^2}]$	$[-0.1\ -0.2\ 0.15]^T$
$b_w[\frac{rad}{s}]$	$[0.01\ 0.02\ -0.015]^T$

After convergence, the filter yield the average results listed in Table 4.3 for the inter-sensor calibration and the visual scale factor L.

The RMS on the estimated pose is for v_w^i [0.044, 0.050, 0.034] m/s in x,y,z respectively and for q_w^i [0.024, 0.022] rad in roll and pitch respectively. Note that we do not list the RMS of the position p_w^i nor the yaw angle of the attitude q_w^i since the observability analysis showed that they are not observable.

Table 4.3: Inter-sensor calibration results on simulated data

	p_i^c[m]	q_i^c rpy[rad]	L	$b_a[m/s^2]$	$b_w[r/s]$
Average error	0.001	0.002		0.037	$< \epsilon$
	0.002	0.003	0.89%	0.033	$< \epsilon$
	0.007	0.005		0.017	$< \epsilon$

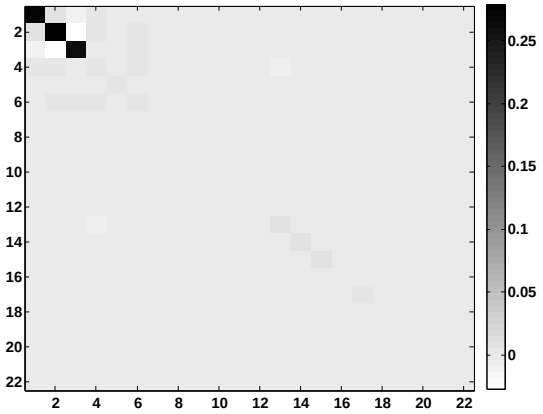


Figure 4.22: State covariance matrix of the converged state. It is clearly visible that the position states p_w^i are not observable and hence they have a large uncertainty (first 3 states in the matrix). Note that the covariance matrix corresponds to the error state which represents all rotations in their minimal form (i.e. 3 elements). Thus the dimension is 22×22 only. (Weiss et al. (2012b))

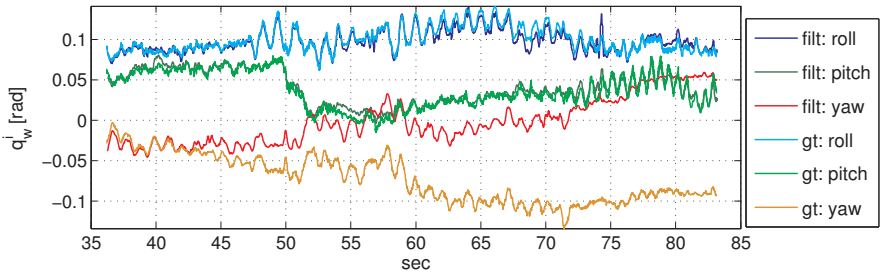


Figure 4.23: Estimated MAV attitude. While roll and pitch are observable the yaw angle is not. This is clearly visible by its drift with respect to the ground truth. The RMS is [0.007, 0.014] rad in roll and pitch, respectively. (Weiss et al. (2012b))

The conducted simulations suggests that the linearization effects do not corrupt the EKF estimation. Fig. 4.22 depicts the state covariance matrix after convergence. The unobservability of the position (first 3 states) is clearly visible by a high uncertainty.

For the experiments on the real platform we used the system explained in Section 4.2.1. As ground truth data, we use a Vicon system with millimeter and sub-degree accuracy. We take the temporal derivative of the Vicon position for ground truth velocity.

We measured the ground truth inter-sensor calibration parameters to be p_i^c = [0.015, -0.01, -0.03] m and rpy(q_i^c) = [0, π, 0]. Since the ground truth scale-factor is very difficult to determine, we omit direct analysis of this state. Instead, the true scale-factor is reflected in the metric ground truth velocity-data. Thus, comparing the velocity estimate of the filter with ground truth implicitly yields a qualitative picture of the correct scale estimate.

Fig. 2.22 also shows that the scale is estimated correctly, since otherwise, the magnitude of the filter's velocity-estimate would differ from the Vicon ground truth. In this experiment, we measured an RMS of [0.028, 0.035, 0.025] m/s in x,y,z, respectively.

Fig. 4.23 illustrates that the roll and pitch angles are observable, while yaw is not. We derived this theoretically in Section 3.4.5. For better comparison, we rotate the angular estimates of our EKF filter to the Vicon ground truth at the beginning. Hence, roll and pitch may not have a zero mean value. The RMS to ground truth is [0.007, 0.014] rad in roll and pitch, respectively.

For the inter-sensor calibration we measured an RMS of [0.009, 0.011, 0.004] rad in roll, pitch, yaw of the attitude between camera and IMU. The results indicate, however, that our ground truth may not be precise enough to judge in detail this RMS value. We experienced the largest issue in estimating

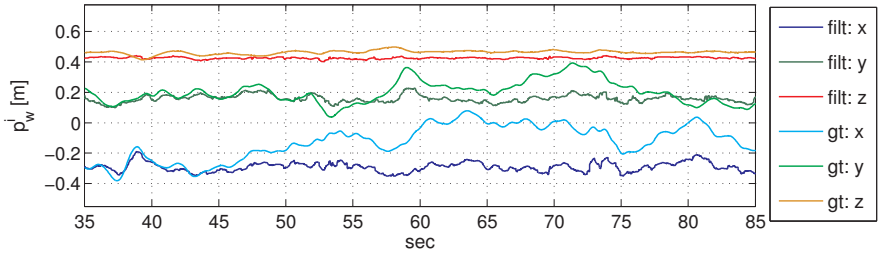


Figure 4.24: Position plot of the autonomously hovering MAV. Because of the non-observability of the position state, it performs a random walk. Notice, however, that this walk is very slow due to the good estimate of the velocity. (Weiss et al. (2012b))

the translation between IMU and camera p_i^c. For this, we measured an RMS of [0.008, 0.005, 0.083] m in x,y,z, respectively. Given the small values for the ground truth distance, this RMS is large. We assume that the system would need more motion in order to converge better. Also, the larger the distance between the sensors, the more relevant is its influence on the measurements and thus the better can it be estimated.

In a new experiment, we let the MAV hover autonomously solely based on our inertial-optical flow approach. Fig. 4.24 shows the position plots. Note that the position performs a slow random walk since it is not observable, however, based on a good velocity estimate. This plot shows, that the MAV is indeed able to hover robustly long enough to perform a (re-)initialization of higher level vision algorithms such as a VSLAM framework.

A sample of the velocity estimate of the filter versus the ground truth is plotted in Fig. 4.25. The fitting in magnitude of the two velocities indicate a correct estimate of the scale factor. The RMS over the whole flight is [0.053, 0.041, 0.017] m/s. We assume that the higher RMS arises from the vibrations while flying and thus possible motion blur and inaccurate feature matching.

For the attitude we measure an RMS of [0.023 0.041] rad for roll and pitch, respectively. After 80 sec of flight, the yaw drifts 0.5 rad. The inter-sensor attitude RMS is [0.045, 0.016, 0.027] rad whereas the inter-sensor distance RMS is [0.033, 0.016, 0.252] m. Again here, we justify the high RMS value for the estimate of p_i^c as due to insufficient movement. In Section 4.1 we saw, that rotational movement in particular would lead to faster state convergence of p_i^c. As we are in hover mode for this experiment, we lack of excitations in angular velocities.

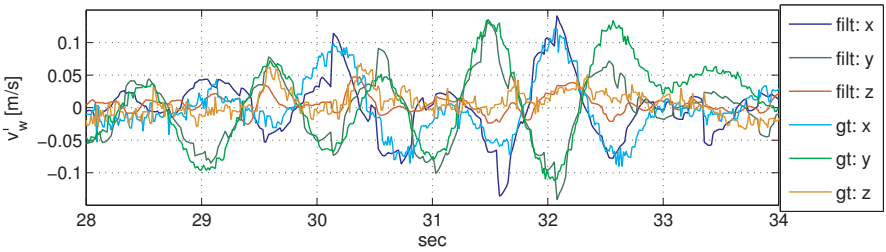


Figure 4.25: Velocity of the autonomously flying MAV. The correct magnitude of the velocity estimate lets us assume that the scale is estimated correctly. We notice also some erroneous vision updates because of too many wrong feature matches (e.g. at sec. 32 in y). The high RMS of [0.053, 0.041, 0.017] m/s may arise from bad feature matching and motion blur in the vision part due to vibrations during flight. (Weiss et al. (2012b))

Initialization of the Improved Monocular Pose Estimator

We showed in Fig. 4.24 that our inertial-optical flow approach is capable of velocity-control the MAV. Thanks to a statistically optimal metric velocity estimation, the position drift is very little (about 20 cm during the whole experiment of 80 sec)

As a further experiment, we initialize our 6DoF position-estimation module to show its capability of re-initializing in case of map-loss. Note that, in this case, we did not use the information of the position-estimation to control the MAV. This was solely done by the previously described inertial-optical flow approach. Fig. 4.26 shows the position estimation from our modified PTAM versus ground truth. After first initialization we scale the visual pose to the ground truth value for easier comparison. At around t=50 s and t=71 s we deliberately canceled the map, which triggered an automatic re-initialization using our improvement on scale and pose propagation. Note that after initialization the position and scale are only slightly different. This is sufficiently accurate to position-control an MAV without having large pose jumps upon re-initialization sequences.

4.3 Results in a Large Environment

In this section, we move our implemented system in Section 4.2 to a large outdoor environment. We will observe how the pure vision based navigation (core-sensor suite) behaves with respect to drifts. Furthermore we highlight

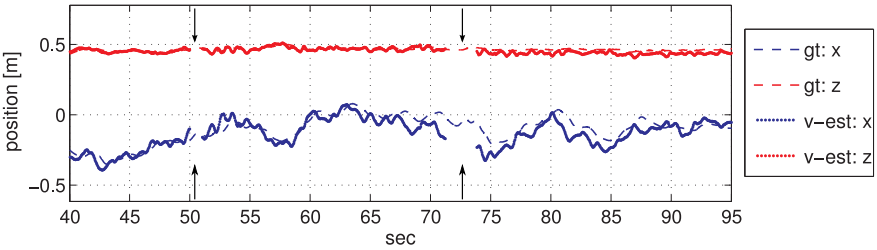


Figure 4.26: (Re-)initialization of our visual 6DoF position estimation. At around t=50 s and t=72 s we force a re-initialization. Note the small change in scale and pose – this would be undefined and arbitrarily large without using our scale and pose propagation improvement discussed in the previous section. (Weiss et al. (2012b))

the addition of a GPS and a magnetometer to the system. Last, we show the behavior on switching from one sensor suite to another during flight. Namely, we initialize our system using the full sensor suite consisting of a camera as a pose sensor, a magnetometer and a GPS receiver. Then we start switching back and forth between vision only navigation and GPS/magnetometer based navigation. We make use of our findings in Section 3.4 to handle the influences of new sensors on the system's state.

The tests presented in this section are of qualitative nature since we lack of precise ground truth (previously provided indoors by a Vicon system). Instead, we use the attitude estimation of Ascending Technologies [7] as ground truth attitude. According to the manufacturer, this attitude is calculated by using the magnetic field and the IMU information in a complementary filter. For position ground truth, we use the raw GPS measurements. The primary testing field was a slightly inclined slope of 40x80 m. The inclination is roughly estimated to be 15 deg.

4.3.1 Vision Based Navigation

In Fig. 4.27 and Fig. 4.28 we show the position and attitude evolution estimated by our filter framework using only the core-sensor suite (camera as a pose sensor and the IMU as a propagation sensor). The position plot shows that the scale has been estimated correctly by the filter and that the position and attitude drifts of the vision system is very low. In Section 3.3 we learned that the position and yaw are only jointly observable with their drift counterparts. That is, for a rapidly drifting vision system we would observe an

[7] www.asctec.com

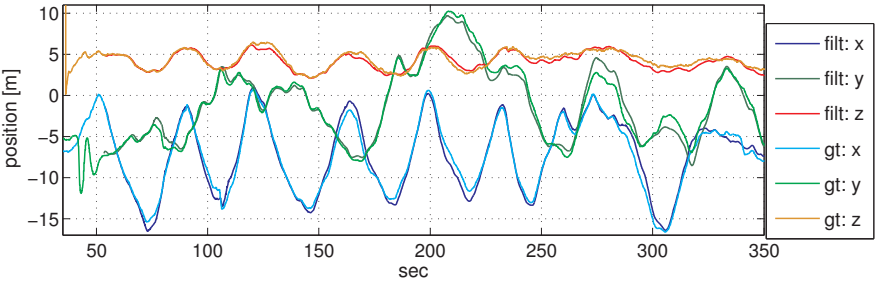


Figure 4.27: Evolution of the position estimate of our filter framework versus raw GPS measurements. The plot suggests that the scale is estimated correctly. Furthermore, the drift of the vision framework in position and yaw seems to be very small since absolute position and yaw are only jointly observable with their drift counterparts.

increasing difference between GPS raw data and filter estimates. In the attitude plot, we see that, although we carefully tried to align the MAV initially with the correct yaw angle, we did not correctly hit the initial filter value in this state. Nevertheless, the filter only needs a short convergence time to find the correct attitude. Note that we used the GPS as additional sensor in the filter to estimate the yaw angle during the initialization phase.

We identified the same issue on the inter-sensor calibration state p_i^c as we observed in our quantitative practical tests in Section 4.2. The distance between IMU and camera seems to be too small for the given low excitations in angular velocities as to estimate this state reliably (see also Fig. 4.3 to gain insight to the convergence behavior of this state). Our estimates were generally about 5 cm off the true value. The error on this state is, however, bounded since large errors would result in large disturbances of other, more robust states. The estimate on the gyroscope bias b_ω and inter-sensor attitude q_i^c were accurately estimated. Since we identified the bias of the accelerometer b_a as a low observable state we plot its evolution in Fig. 4.29. Even though the estimate is noisy and slow converging, it tends towards the correct values.

The vision attitude offset in roll and pitch with respect our reference frame has been estimated to 4.4 deg in roll and 15.8 deg in pitch which is a reasonable value for our inclined test area. The evolution of the drift is plotted in Fig. 4.30 in degrees. We assume the drift to be very low because of the well distributed features outdoors and the fact that we did not have any map losses while flying. This shows that our improvements on the visual pose estimator not only render it capable of running in real time on-board

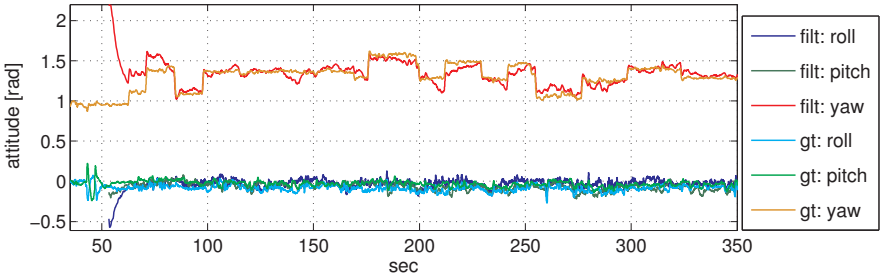


Figure 4.28: Evolution of the attitude estimate of our filter framework versus the attitude estimate done by Ascending Technologies. In contrast to the position, we cannot initialize the attitude very precisely. The plot shows that we initialized it off the true value. Only after a short initialization time, the filter converged to the correct attitude. (initial yaw angle was estimated by adding a GPS sensor)

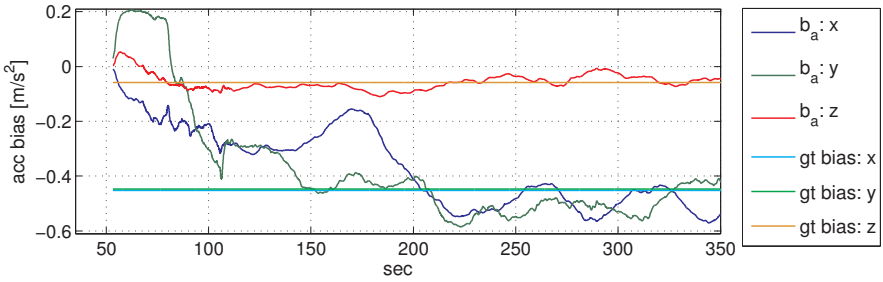


Figure 4.29: Evolution of the acceleration bias estimate of our filter framework versus the mean value over the measured accelerations measured during the whole flight used as a ground truth. The states slowly converge to a reasonable value.

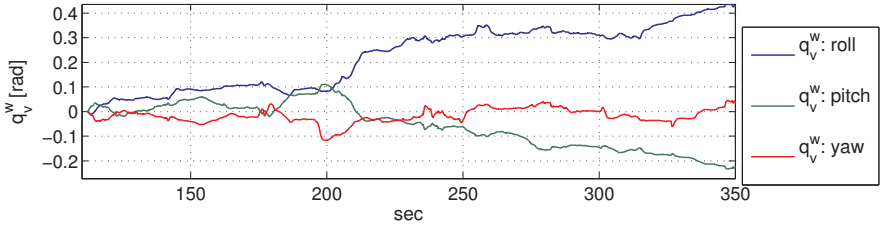


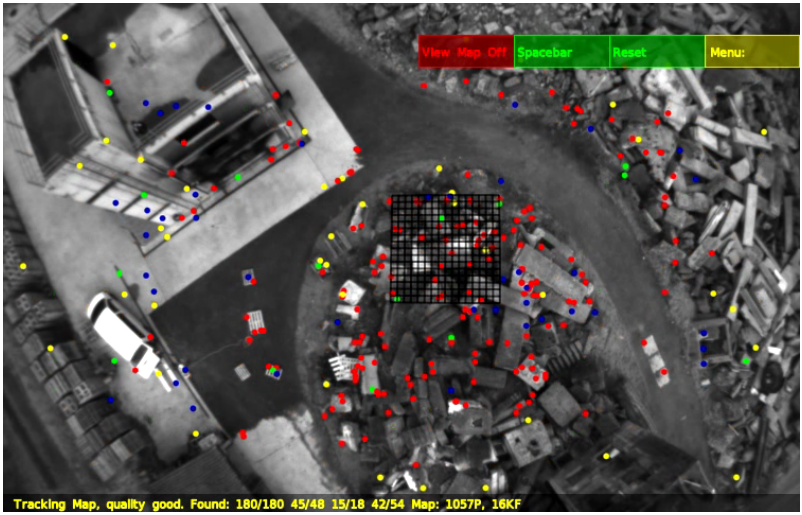
Figure 4.30: Evolution of the visual drift in roll and pitch with respect to our world reference frame in degrees. We assume that the very low drift is a result of the good feature distributions outdoors and the fact that we did not have any map losses during flight.

the MAV but also render it sufficiently robust for outdoor scenarios. For this flight we used a maximum of 5 key-frames.

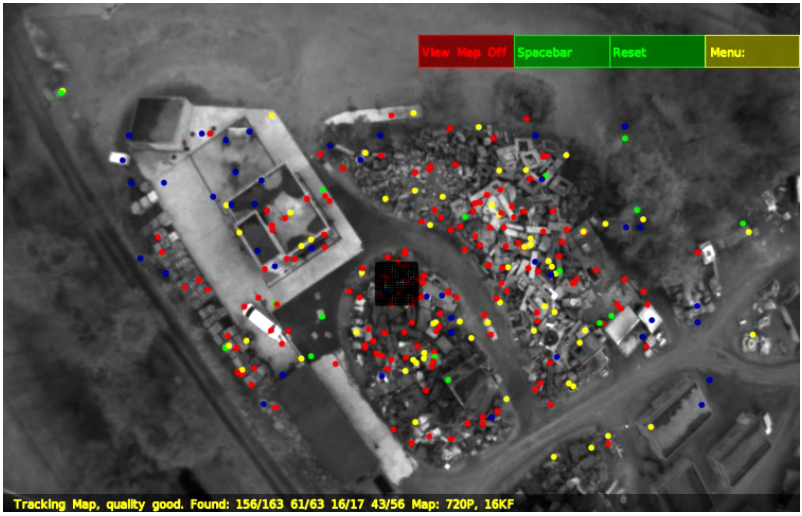
We also tested the visual based navigation system on height changes under windy outdoor conditions. For these tests we used a new testing area which allowed such maneuvers. Fig. 4.31 depicts the area showing a disaster-like environment. We do not discuss all our experiments but show a representative height test in Fig. 4.33. There, we plot the 3D position to show the capability of our system of navigating up to 70 m above ground and then land autonomously in the same mission. Fig. 4.32 shows the camera view attached to the helicopter at a height of 70 m. During the whole flight we only used our core sensor suite, i.e. one single camera and an IMU as sensors. This height change could not be tackled by a stereo-setup und demonstrates the strength and benefits of our vision based navigation framework described in this dissertation. In Fig. 4.34(a) the hovering performance at 50 m above ground and during wind disturbances is shown. In Fig. 4.34(b) we plot the attitude during the hovering. The offset in roll is not a wrong attitude estimate of our framework but shows that the MAV acts actively against wind in order to keep its position. To our knowledge, navigating within this height range (i.e. from 0 m to 70 m) in the same mission and at this accuracy despite wind disturbances is unmatched in vision based navigation for micro helicopters.

In the same testing area, we performed very long-term vision based navigation using our core sensor suite only (one camera and one IMU). A representative long-term navigation test is depicted in Fig. 4.35 where we show the trajectory flown for one battery lifetime. The maximum speed was 2 m/s and the overall trajectory was about 360 m. The loops were flown by re-

Figure 4.31: Testing area for altitude and long-term navigation tests. The area is a firefighter training area and resembles a common disaster area after an earth quake for example. (Bing Maps)

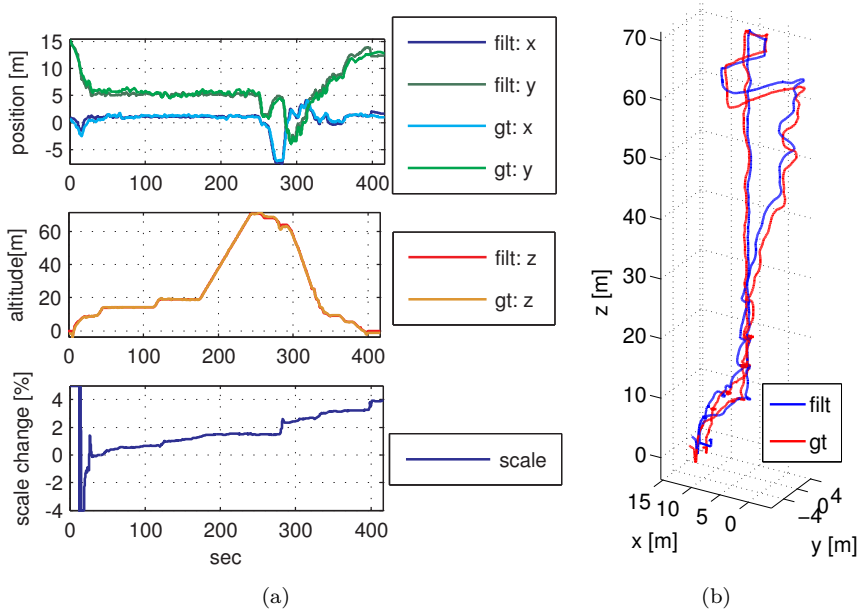


(a)



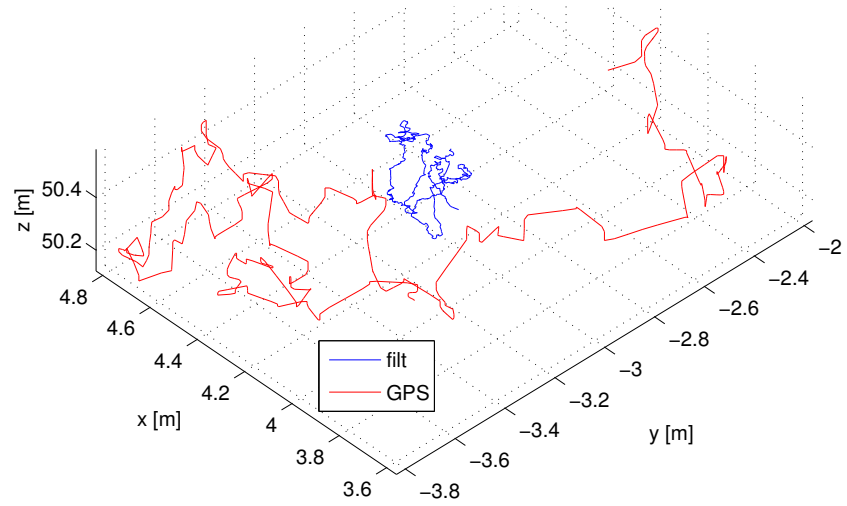
(b)

Figure 4.32: Camera view with overlaid features and map grid while navigating purely vision based first to a) 35 m and ascending to b) 70 m above ground.

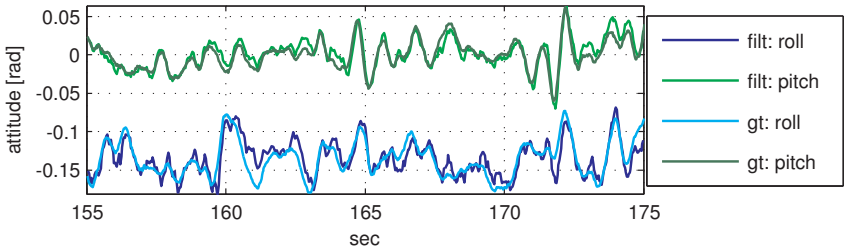


(a) (b)

Figure 4.33: Altitude test: until t=30 s we perform an initialization phase using GPS and the visual pose estimate as measurements to converge all states in the framework. After t=30 s we switch to vision only navigation (i.e. we use only one camera and an IMU as sensors) and climb up to 70 m without intended lateral motion. Then we let the helicopter descend while applying lateral motion to show the maneuverability during this drastic height change. The end of the graphs show safe vision based landing still using only one camera and an IMU. The lower plot in a) depicts the evolution of the visual scale λ – after initialization (t=30 s) it slowly drifts as expected.



(a)



(b)

Figure 4.34: a) Hover performance at an altitude of 50 m above ground during 20 s. We see that our vision based framework still yields a much better pose estimate than the GPS signal at this altitude (the scale was correctly estimated by our framework). This hovering test was performed during wind. b) shows the roll offset of the helicopter's attitude in order to counteract the wind. Note that also the attitude estimate is still correct at this altitude.

peatedly sending way-points. In the third loop the battery run low such that the helicopter had to land. Fig. 4.36 shows good overlap of the 3D position estimation of the filter with the GPS ground truth indicating not only a correct estimate of the drifting scale but also indicating that the position and yaw drift of our improved onboard visual odometry framework is very small. In fact, we measured a position drift at the end of the flight of only 1.47 m which corresponds to about 0.4%.

The visual attitude drift in roll and pitch as well as the visual scale drift are depicted in Fig. 4.36 in the lower plots. Note that we do not apply any sort of active loop closure and only keep the last 15 key-frames (about 10 m at a height of 13 m above ground) with their associated features in the local map. To our knowledge, this is to date the longest path flown by a micro helicopter based on purely vision based navigation. We note that the limiting factor in these experiments was the battery lifetime, not a part of our framework.

4.3.2 Multi-Sensor Navigation

In order to eliminate the drifts according to our findings in Section 3.4 we added a magnetometer and a GPS sensor to the core-sensor suite. The following tests were performed only in the primary testing area described before. In Fig. 4.28 we already showed the capability of the system finding the correct yaw angle. Since we have a magnetometer *and* a GPS sensor we can estimate both the elevation angle α and azimuth angle β of the magnetic field vector in the world reference frame. The framework estimated the elevation angle to be $\alpha = -65.7$ deg. At the present location, the magnetic inclination is -63.3 deg resulting in an error of the estimate of 2.4 deg. The GPS receiver of Ascending Technologies operates in a coordinate system which is 90 deg rotated to the geographic north (i.e. East-North-Up system). The azimuth angle has been estimated to be $\beta = 91.9$ deg. Considering the magnetic declination at the place of the testing area, this is an error of 3.2 deg. We note, that the magnetic field in the test area could be distorted by the surrounding installations. The position offset between the vision frame and the world reference frame (i.e. the GPS zero point set at the beginning of the experiment) has been estimated to be $p_v^w[8.4, 9, -2.3]^T$ m which lies in the accuracy of our hand measured values. The negative value in z arises from our map initialization position being further up the slope than the GPS zero reference point.

We used the full sensor suite (camera as a pose sensor, magnetometer, GPS) to show the switching between different sensor modalities. From Sec-

Figure 4.35: Trajectory flown in the disaster area. After a short initialization phase where we used GPS and the camera as exteroceptive sensors in order to converge all states of the framework, we switched to purely vision based navigation until the battery run low and the helicopter had to land. The total length of the trajectory navigated only vision based is about 360 m. The good overlap with the GPS ground truth indicate a very low position and yaw drift of our improved real-time and onboard visual odometry framework.

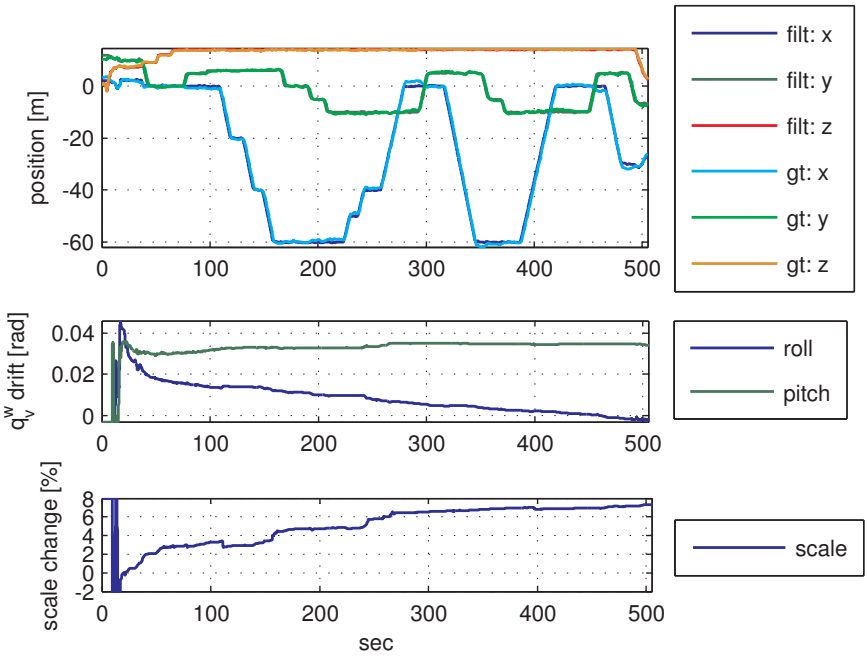


Figure 4.36: Top graph: shows the trajectory flown and depicted in Fig. 4.35. The heights was kept constant to 13 m. The saddle points in the graph reflect hovering between sending waypoints. Middle plot: shows the visual attitude drift while flying purely vision based. We only show the evolution of the roll and pitch drift since the yaw is unobservable in this sensor configuration. The estimation of this drift is crucial in order to keep the helicopter aligned with gravity and thus to prevent a crash in long-term missions. Lower plot: shows the evolution of the visual scale factor. The continuous estimate of this state ensures accurate navigation and control in a metric coordinate frame.

tion 3.4 we know that the GPS sensor-setup renders all involved states observable. Thus, we may simply add the camera and wait for convergence of the newly introduced states. Then, we switch to vision only navigation. In this setup, we force a map reset causing the filter framework to switch back to GPS and magnetometer based navigation until the states introduced by adding a camera are converged again. Then we switch a last time to vision only navigation.

Fig. 4.37, Fig. 4.38 and Fig. 4.39 show the state evolution of the position and attitude drift states p_v^w and q_v^w as well as the visual scale state λ. At t=13 s we start our filter framework and navigate with GPS, magnetometer, and camera. At t=50 s the states sufficiently converged to switch to vision only navigation. Fig. 4.39 shows that the scale is still reliably estimated. This is also reflected in the accurate position estimate depicted in Fig. 4.40. We force a map reset at t=162 s and switch to GPS and magnetometer navigation until t=190 s. The switching is manifested in a small position-jump. Such position-jumps would be fatal for an underlying controller if they were undetectable. However, a map loss and the linked change in the available sensors is well detectable and can be handled accordingly by a controller. This is in particular true since the estimates after the position-jump are again consistent. A similar but even smaller jump in yaw can be observed in the MAV attitude evolution plot Fig. 4.41 at t=162 s. The jumps are relatively small since the drift of the visual framework is nearly negligible in this data-set.

At t=190 s, we initialize the map again at the new current MAV location (no scale and pose propagation is used here). We wait until the camera drift states are converged again to the new map location. Then, we switch the framework back to vision only navigation at t=213 s

The position of the new map is further in the positive y and negative x axis (downhill) which also means closer to the GPS origin in z direction. This is represented in the decreased offset p_v^w between the freshly initialized vision frame and the world reference frame. The filter converges to a reasonable value. For the attitude drift state q_v^w in Fig. 4.41, it is very difficult to judge its correctness. However, given the inclined testing area and the fact, that the MAV kept a similar yaw angle for both map initializations, the values are reasonable.

Finally, in Fig. 4.42 we show a part of the 3D path estimated by our filter and compared to the raw GPS measurements. The part corresponds to the section between t=147 s and t=222 s where we forced the map reset. At the instant when we change to GPS and magnetometer navigation the position estimate experiences a jump. This reflects the visual position drift until this

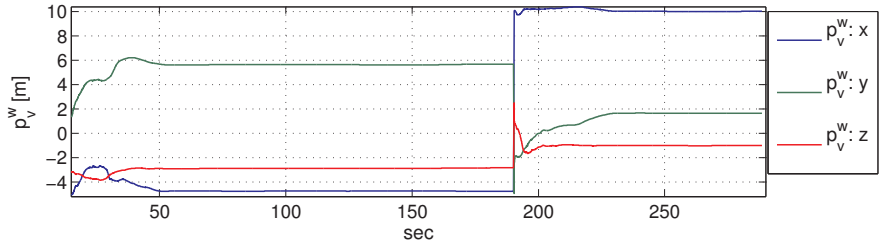


Figure 4.37: Evolution of the visual drift in position. At t=162 s we force a map reset and initialize it at a new position at t=190 s. The drift (or offset) state p_v^w between world reference frame and vision frame gets reasonably estimated after some convergence time. Without our improvement of propagating the map pose upon re-initialization, the new map may lie at an arbitrary position.

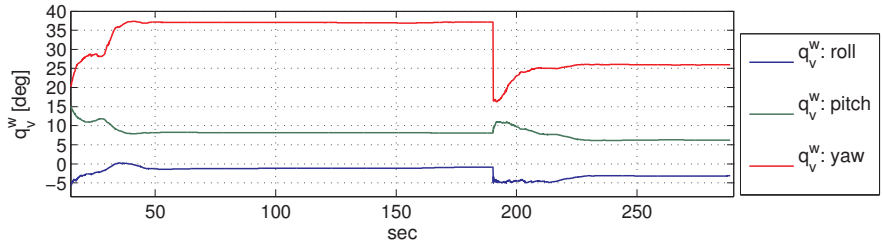


Figure 4.38: Evolution of the visual drift in attitude. At t=162 s we force a map reset and initialize it at a new position at t=190 s. While roll and pitch of the offset remain the same as we are on the same slope, the yaw changed slightly since the MAV had a different heading with respect to the first map initialization.

point in time. In our case this is about 1 m in z and y and negligible in x direction. The position marked with [*] defines the point where we changed from vision only navigation to GPS and magnetometer navigation. [o] marks the point when we added the camera again with a new map and [+] marks the point where we switched back to vision only navigation.

Fig. 4.43, Fig. 4.44 and Fig. 4.45 show an forced map reset while using our scale and pose propagation improvement. All map properties p_v^w, q_v^w and λ remained the same after re-initialization upon map loss. Only the y value of p_v^w has a difference of about 1 m. The reason of this offset the distance traveled between detecting the map failure and re-initializing the new map.

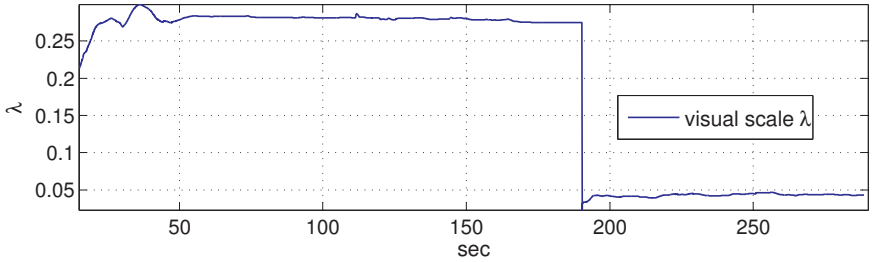


Figure 4.39: Evolution of the visual scale. At t=162 s we force a map reset and initialize it at a new position at t=190 s. Without our improvement of propagating the map scale upon re-initialization, the new map may have an arbitrary scale.

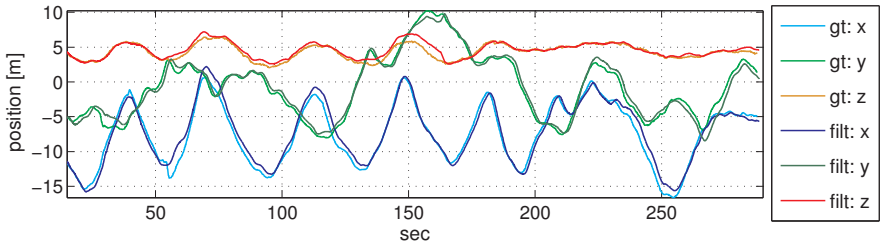


Figure 4.40: Evolution of the MAV position compared to the raw GPS measurements. Note the correction in position after the map reset at t=162 s. The estimator is forced to switch to GPS and magnetometer based navigation mode and corrects the drifted vision position to the GPS readings. A map loss/reset can be detected easily and handled accordingly in the position controller for the MAV. The position-jump only occurs on at the time of switching *to* GPS mode since this sensor setup renders some unobservable states in the vision only setup to observable states.

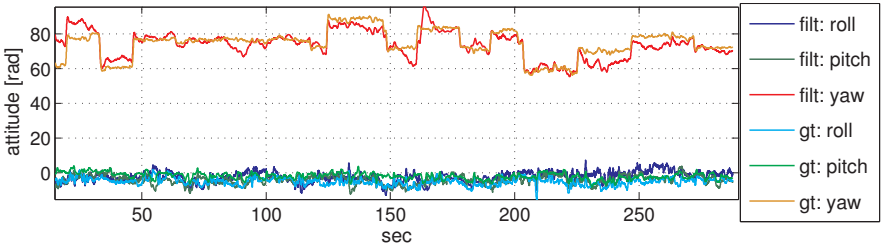


Figure 4.41: Evolution of the MAV position compared to the raw GPS measurements. Note the correction in position after the map reset at t=162 s. The estimator is forced to switch to GPS and magnetometer based navigation mode and corrects the drifted vision yaw to the GPS readings. A map loss/reset can be detected easily and handled accordingly in the position and attitude controller for the MAV. The jump only occurs on at the time of switching *to* GPS mode.

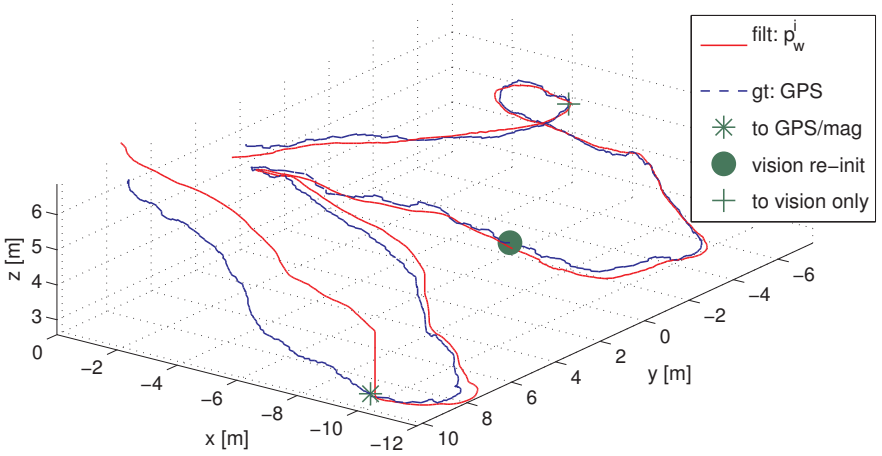


Figure 4.42: 3D path of a part of the data-set. The part depicts the trajectory between t=195 s and t=255 s. We see the foced re-initialization step (marker [*]) causing the estimator to switch to GPS and magnetometer mode. The position-jump upon the switching is caused by the position drift while navigating with vision only. The marker [o] shows where we re-initialized the map and where we switched to GPS, magnetometer and vision based navigation. After convergence of states introduced by the camera addition, we switch to vision only navigation at marker [+].

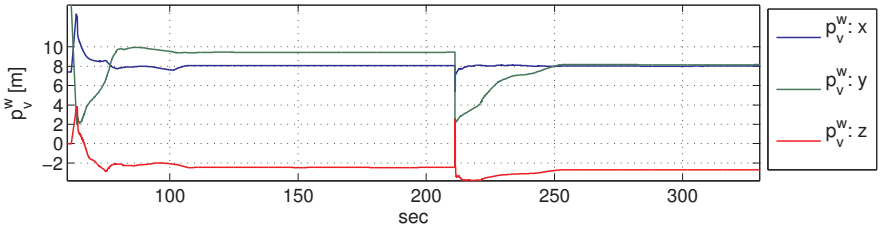


Figure 4.43: Evolution of the visual drift in position. At t=75 s we force a map reset and re-initialization at t=211 s using our pose propagation improvement. The drift (or offset) state p_v^w between world reference frame and vision frame gets reasonably estimated after some convergence time and is well propagation such that it remains unchanged even after map re-initialization. Except the y part changed by about 1 m. This is because the movement in y between map-loss detection and re-initialization.

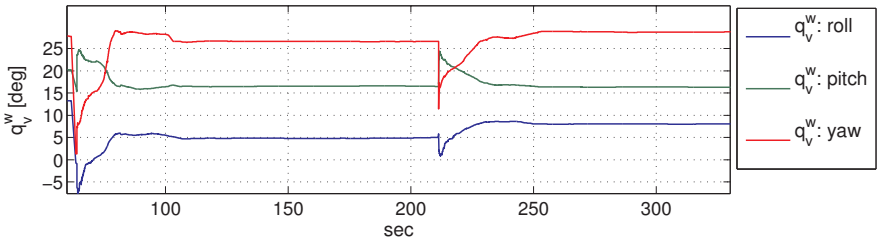


Figure 4.44: Evolution of the visual drift in attitude. At t=75 s we add the camera as a pose sensor. Since all states in the GPS-IMU setup are already observable, we can simply add the camera and wait a short time for convergence of the additional states (q_v^w in this plot). At t=211 s we force a map reset and re-initialization using our pose propagation improvement. The negligible change in the map's attitude with respect to the world reference frame shows well functioning of our map pose propagation improvement upon map re-initialization.

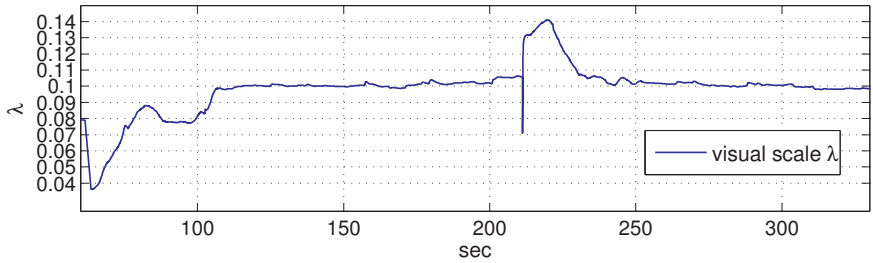


Figure 4.45: Evolution of the visual scale. At t=75 s we add the camera as a pose sensor. Since all states in the GPS-IMU setup are already observable, we can simply add the camera and wait a short time for convergence of the additional states (λ in this plot). At t=211 s we force a map reset and re-initialization using our pose propagation improvement. The well functioning of scale propagation upon re-initialization is evident in the plot.

4.4 Conclusion

We carefully implemented and analyzed the sensor fusion filter based on the theory discussed in Chapter 3 and evaluated its sensitivities to initial state errors. We also demonstrated successfully the functioning of the filter on real data and showed that we reliably can identify failures and drifts of the pose-estimation algorithm. This ensures proper functioning of the filter only having access to the pose output of the black box.

Since our observability analysis on the non-linear system only is a necessary but not sufficient criteria for the linearized and discretized system, we perform extensive experiments and on both simulated and real data. These experiments reveal the dependency of successful filter convergence on a good initialization of the visual scale factor. Even though the visual scale factor is a delicate state for initialization, we show that our system is able to precisely track variations of this and other drift factors of the visual pose estimator.

Driven by the need for truly autonomous and robust navigation of MAVs, this chapter presents a system for online inter-sensor calibration and 6DoF pose estimation in an distributed implementation that runs entirely on-board a MAV (with an ATOM 1.6 GHz processor and ARM7 processor). Our results demonstrate the power and modularity of our framework using different sensors in addition to an IMU. The distributed EKF architecture we propose allows a MAV state prediction at 1 kHz for accurate control, while time-delays are compensated such that exact filter updates are ensured in a way

that computational load is optimally distributed across updates. Our thorough quantitative evaluation shows that our approach is highly robust against noise, delay and slow measurement readings.

Exploiting the full potential of the visual-inertial sensor fusion, we demonstrate both the metric 6DoF pose estimate and inter-sensor calibration with all processing on-board the MAV and in real-time.

For situations where the visual pose estimator fails, we showed results on the performance of our map-less velocity control approach based on inertial optical flow. We demonstrated, that thanks to the high-quality velocity estimation the framework only drifts very slightly in position leaving sufficient time for re-initialization procedures for map-based approaches. Also, for the map-less approach we showed good behavior of the (inter-)sensor state convergence rendering also this system into a power-on-and-go platform.

In addition to the extensive quantitative evaluation of our distributed implementation, we tested the framework outdoors in a large real-world environment. These tests prove the functioning of our approach in real long-term navigation scenarios for real power-on-and-go systems. More precisely, we showed that the estimated pose, using only vision-inertial navigation, fitted well the raw GPS measurements and that all (inter-)sensor states were estimated correctly. A second test series showed the proper switching between different sensor suites. We applied our previous findings to determine the effects of different sensors on the vehicle's state space and handled the switching to different sensor modalities accordingly. We analyzed different transitions from the full sensor suite (camera, GPS, magnetometer) to vision only to GPS and magnetometer only and back to vision only navigation. Each transition was smooth and the newly added states converged to the correct values.

As a summary, in this chapter we proved the proper functioning of long-term vision based navigation in a large and real-world environment. Our system can handle different sensor-modalities in flight, is fully self-calibrating and robust to failure modes in the visual pose estimator, thanks to a map-less body-velocity estimator.

Chapter 5

Discussion and Conclusion

5.1 Summary of this Dissertation

In this dissertation we examined a versatile approach for long-term vision-based navigation for micro helicopters in large environments. That is, we propose a method which primarily uses an IMU and a camera for MAV navigation and provide detailed insight on how to add different sensors to mitigate drift issues. For vision based navigation, we first defined the camera as a valid real-time and on-board 6DoF pose or 3DoF body velocity sensor in Chapter 2. The fact that we can treat the camera as a regular motion sensor was then used in our theoretical analysis on optimal sensor fusion for power-on-and-go systems in Chapter 3.

Our findings provide a detailed overview on which sensors render which vehicle and (inter-)sensor calibration states observable such that we can find a suitable sensor suite for different tasks. Our practical tests in Chapter 4 prove the correctness of our theoretical findings and show that the theory derived can also be used to smoothly switch from one sensor suite to another during flight.

5.2 Discussion of Contributions

The findings and results presented in this dissertation are a step towards vision based navigation for agile, airborne power-on-and-go systems. In this chapter, we summarize the results and put them into perspective of future work. For the discussion, we follow the tasks sketched in the introduction of this document: first, we overview the achievements on having the camera as

a real-time on-board motion sensor. Second, we summarize our theoretical findings on the modular multi-sensor system we propose. Last we recall the viability of our theoretical findings in simulations and real world tests.

5.2.1 Camera as a Motion Sensor

Many of the state-of-the-art approaches consider the camera as a bearing sensor able to detect salient points in the environments (i.e. features). Camera-motion estimation algorithms based on this assumption typically are computationally very heavy and are not applicable on computationally constrained platforms. We proposed in Chapter 2 a two-fold contribution on the use of the camera as an abstracted sensor yielding a 6DoF pose (map-based approach) or a 3DoF body velocity (map-less approach).

As for the abstraction of the camera to a 6DoF pose sensor – we referred to this as the map-based approach – we modified a high-performant state-of-the-art key-frame-based monocular SLAM framework (Klein and Murray (2007)) to a visual odometry framework capable of running at 20 Hz on an on-board ATOM 1.6 GHz single core platform. To our knowledge, our modifications changed the original framework – meant to be used in small indoor scenarios only – into the fastest visual odometry framework sufficiently robust for large-scale outdoor tasks. The abstraction from a bearing sensor to a 6DoF pose sensor brought different advantages and disadvantages to the overall system. While approaches using the camera as a bearing sensor (i.e. EKF SLAM (Kelly and Sukhatme (2011))) inherently tackle the frame in which the features are observed as reference frame, we are independent of such an inherent definition. More precisely, by decoupling the camera to be a standalone 6DoF pose sensor we can actively introduce a world reference frame additionally to the camera's own reference frame. The world reference-frame is aligned with gravity to ensure proper MAV navigation. This reveals the so-called drift states between the vision frame and the world reference frame. These states are hidden in approaches using the camera as a bearing sensor and the system's reference frame inherently drifts along the system's unobservable drift directions. As we treat these drift states explicitly, our approach yields precise knowledge about the system's drifts and unobservable modes. Furthermore, the separate world reference frame allows now to add any additional (and in particular drift-compensating) sensors and turns our approach into a versatile multi-sensor system. Being aware of the drift states and the unobservability of the overall system, our contribution contains an extensive analysis of the behavior of our improved visual pose estimation algorithm. This gives deep insight to the motivation and effects

of our modifications.

As for the abstraction to a 3DoF body velocity – we referred to this as the map-less approach – we provide a contribution to recover the scaled camera velocity using inertial constraints. More precisely, we use the angular velocities of the IMU measurements to un-rotate the measured optical-flow vectors. These vectors provide a measure of the camera velocity in 3D. The approach yields the camera velocity at 40 Hz on an ATOM 1.6 GHz single core processor board using a standard WVGA camera. To our knowledge, this approach is unique in the sense that we simplified the continuous 8-point algorithm (Ma et al. (2000)) down to 2 dimensions: the direction of the 3D camera velocity. This has a couple of advantages. First, we only need a minimum of two feature matches in two consecutive images in order to calculate the camera velocity. This means, we are not bound to a feature history nor to a map. Thus, a failing of the algorithm is highly unlikely given sufficient texture in the scene. Second, since only two points are sufficient for the velocity recovery, RANSAC approaches for outlier rejections are extremely fast and robust. Third, the approach is robust against zero displacement between two camera images (in contrast to the discrete version which needs an appropriate baseline for proper functioning).

One might argue that a scaled body velocity is insufficient for proper MAV velocity control. However, using the theory provided in Section 3.4.5 we showed in Section 4.2.7 that we can not only control the MAV accurately in metric velocity but also are able to estimate the scale factor, the IMU biases and all inter-sensor calibration states. This turns the abstraction of the camera as a 3D body velocity sensor into a powerful approach for a barely failing velocity-controlled power-on-and-go system. To our knowledge, this is the first self-calibrating inertial-optical flow method used for accurate onboard velocity control of an MAV. Furthermore, with this approach we have a stable fallback for any more sophisticated map based visual pose methods used for position control the vehicle.

5.2.2 Multi-Sensor Power-on-and-go-Systems

While the abstraction of the camera to a general motion sensor is a necessary step for modular sensor-fusion, our main contribution is the actual proposition of such a modular sensor-fusion method as well as its detailed theoretical analysis and testing. Our approach differs from the state-of-the-art in the sense that we not only aim at a pose estimate for proper MAV control but also and at the same time we aim at the (inter-)sensor calibration of any sensor suite used for the current task. Furthermore, our approach is modular,

runs in real-time and on-board the vehicle. It is also inherently suitable for long-term navigation because of the continuous self-calibration of the sensor suite.

The ability of self-calibration not only eliminates the need of prior and tedious calibration procedures but also enables the MAV to switch from one sensor suite to another *while flying*.

Theoretical Analysis

For our approach, we decoupled the frame in which the camera perceives features from a world reference frame, which allows treating the visual pose estimator as a black box yielding an unscaled 6DoF pose. This renders our approach independent of the type of visual pose estimator. This allows using less computationally expensive key-frame based structure from motion or visual odometry solutions, as we are not bound anymore to the computationally expensive EKF based visual pose estimators. Visual odometry approaches can be implemented on platforms with very limited calculation power and are not bound to a finite working space. This makes our approach scalable to robot navigation in large environments even for platform with low calculation power. Furthermore, the introduction of a separate world reference frame gives a versatile basis for further sensor additions.

The possible addition (and removal) of different sensor modalities requires a deep understanding of the system in each configuration. Already the bespoken drift states introduced when using the camera as a separate 6DoF pose sensors are crucial to understand in detail. A simple thought experiment reveals this importance: the analysis of Section 3.3 shows that the global attitude (with respect to the gravity aligned world reference frame) and the attitude drift of the vision frame are only jointly observable. If our analysis would reveal the joint observability between these two states did *not* imply that the global roll and pitch angle are directly observable, our system would not fly for long but soon crash because of the misalignment to gravity.

Treating different sensors and in particular the pose estimator as a black box requires the handling of failure and drift estimate issues. We applied a non-linear observability analysis and showed that even with our core setup, consisting of an IMU and single camera only, we are able to navigate drift free in roll and pitch through large environments. In a second step we added further sensors to our core setup in order to eliminate different drift states. With the help of an observability analysis we isolated drifting terms and provided examples for further inclusion of additional drift-compensating sensors, such as a visual compass, a three-axis magnetometer, a sun sensor, or a GPS receiver.

While the sensors GPS and magnetometer present obvious additions to our core sensor suite consisting of one camera and an IMU, we may as well consider them as representatives of a sensor class yielding a specific measurement *type*. In the case of the GPS we can consider the *type* as a global 3-axis position sensor without any drift components. The Magnetometer would then be a type of a drift-free global direction-measurement. In Section 3.4.6 we went one level of abstraction further and considered virtual single axis sensors in global position, global velocity and body velocity as well as attitude and direction sensors. For each such *type* we considered the most general case by adding possible drift components. As an example, a 1D global position sensor with drift in attitude and position would correspond to one axis of the position measurement when the camera is used as a pose sensor.

We then analyzed each of this virtual sensor with respect to its contributions on directly and jointly observable modes to the system. This analysis provided information on the observable states on one side and on the unobservable states and their inter-connectivity with different continuous symmetries on the other side. We listed the summary of these findings in Fig. 3.12. This list provides a powerful tool to combine different sensors in order to render specific states observable. In addition it provides insight to which sensor affects with MAV states. This provides a foundation for sensor switching strategies in flight as it is possible to perform an adapted initialization to maintain the flight stability of the MAV.As a result, we have a theoretically founded tool to construct any real-world sensor and directly obtain the observable and jointly observable subspace in the state space.

We finally gave a theoretical intuition on the quality of the observability of a specific state by analyzing the non-linear time continuous system. We made use of the fact that we only analyze locally weakly observability of the system and that we can approximate the impact of one vector field along a flux of another by a Taylor series. We then used the convergence of this series to define that dimensions in the state space spanned with higher order Taylor coefficients are *less* observable than those spanned by lower order Taylor coefficients. We define *less* observable as being sensitive to noise levels, whereas *well* observable states even yield good convergence (i.e. small RMSE) with large noise values. To the best of our knowledge, this is the first work considering the quality of observability in this theoretical construct of differential geometry.

Practical Implementation

The theory provided in Chapter 3 gives a solid foundation for the observability of a non-linear, time continuous multi-sensor system for long-term MAV

navigation. However, for the practical implementation the validity of the observability analysis has to be shown. Compared to the theoretical non-linear and continuous system, real world system suffer from error sources like linearization and discretization on one side and modeling and time synchronization errors on the other side. Further, the long-term navigation in large unknown environments has to be investigated. In Chapter 4 we therefore provide extensive tests both in simulation and on the real platform.

Section 4.1 shows the validity of the linearization and discretization of the previously analyzed theoretical system. In simulation we put a special focus on the initialization procedure and different levels of excitations of the system, as an initialization of a state off its true value shows the effects of linearization of the system. Initialization errors could have the effects that the system linearizes around a wrong working point and may thus converge to a local minimum only. With the help of our simulations we identified mainly two states which are sensitive to wrong initialization values. First, the scale factor and second the gyroscope biases. Fortunately, the latter ones are observable if the MAV stands still (e.g. on the ground before takeoff). Hence, we can assure an initialization value close to the correct one. The initialization of a correct scale factor is more difficult. We do not discuss details in this dissertation but refer to our method presented in (Kneip et al. (2011)). The remaining states showed to be less critical in wrong initialization values.

Whereas linearization and discretization are one part of the error sources compared with the theoretical non-linear and continuous system, modeling errors and time synchronization issues are the other part. In particular, real-world systems and sensors usually do not have perfect Gaussian noise as applied in the simulation steps. Furthermore, because of the processing time needed, visual pose estimates are always delayed with respect to IMU (and most other) readings. In Section 4.2 we presented a framework which is capable to handle significant time delays and noise levels. Practical tests on the real flying system showed stable flight even with sensor signals having a delay of 500 ms or noise with a standard deviation of $\sigma_p = 20$ cm in position and $\sigma_q = 2\circ$ in attitude. A distributed implementation of the analyzed sensor-fusion approach on-board the vehicle led to a MAV pose and velocity control at 1 kHz with a simple and standard PID controller.

After the successful tests on linearization, discretization, modeling and timing issues we tested in Section 4.3 long-term navigation in large and unknown environments on the real system outdoors. Based on the theoretical findings in Chapter 3 on which sensors influence which dimensions in the state space we presented different scenarios. First, we showed that vision

only [1] navigation is possible as long as we are only concerned about the local consistency of our environment (i.e. position and global yaw drift can be neglected for the task in question). As a practical example, we showed a long-distance outdoor flight of about 360 m with 0.4%drift and an altitude flight between 0 m and 70 m height (including landing). Second, we demonstrated the full capabilities of our modular sensor-fusion approach by switching between different sensor setups *in flight*. Given the theory, for each addition and removal of a specific sensor, we know exactly its influence on the observability of the state space. We tested the switching between the different sensor setups we discussed, in order to analyze the states which are affected by the switching. These tests showed that when switching from GPS only navigation to vision only navigation, we have to wait until the newly introduced states by the camera are converged, however, the full MAV state (including global yaw) is observable. A switching strategy on the controller side is thus not necessary. On the other hand, when switching from vision only to GPS only navigation, we know that the position as well as the global yaw might suffer sudden changes and a switching strategy on the controller side has to be implemented (i.e. ignore the state jumps and use the new values as new reference). Conversely, global roll and pitch do not need to be considered since they are in both setups observable. With these considerations we showed stable flight switching from one sensor suite to the other in flight. These tests in Section 4.3 demonstrate long-term multi-sensor MAV navigation in unknown environments with the focus on vision based navigation methods.

5.3 Applications and Research Outlook

This dissertation presents a basis towards power-on-and-go systems with the focus on long-term navigation of MAVs in unknown environment using primarily vision based methods. While we present an incremental step to the final goal of having aerial vehicles performing autonomously complex tasks, the way to this goal remains long. This section is dedicated to some thoughts on both, applications already applied to our method presented in this dissertation and applications and research directions yet to be accomplished.

[1] we consider the IMU as a fix sensor used for model propagation. Vision *only* navigation is thus referred to the measurement sensors and means in this case that we use an IMU and a camera as only sensors.

5.3.1 Area Mapping and Coverage

This dissertation focuses on the low level task of maintaining a micro helicopter airborne for long periods of time in unknown environments. Once we consider this task as sufficiently tackled, it is tempting to go a step further to autonomous exploration and surveillance using such MAVs. Evidently, for such tasks, some sort of path planning and thus notion of the underlying environment is crucial. In (Weiss et al. (2011a)) we present a lightweight approach to build a textured elevation mesh map of the environment on-board and in rel-time. Aid of the 3D point cloud we obtain from the here presented vision based approach using the camera as a pose sensor we generate an elevation mesh map of the environment. That is, the point information is already calculated and is available without further computation cost. An adapted and improved Delaunay triangulation procedure ensures the real-time aspect of the elevation mesh. Not only have we the 3D point cloud from the visual pose estimator but we have also the camera images at the key-frame locations. That is we can texture the meshed elevation map accordingly and provide a user with semantic information about the environment. This allows inspection tasks with human interactions. The basic steps of the approach presented in (Weiss et al. (2011a)) are depicted in Fig. 5.1

Since we are now able to navigate in large environments, this method can also be applied outdoors for large areas and provide the user with a good overview of the situation and the environment. Fig. 5.3 depicts such a textured elevation map of the area shown in Fig. 5.2.

For pure obstacle and path planning purpose a mesh map without texture would be sufficient. As we can build such maps on the flight in real-time, we can apply incremental coverage and surveillance algorithms also to a multitude of MAVs. We followed this approach in our preliminary studies in (Doitsidis et al. (2011)). A sample mesh map for obstacle avoidance and coverage is displayed in Fig. 5.5(a) and Fig. 5.5(b). The area covered in these maps is displayed in Fig. 5.4(a). In this figure, We also superimposed the trajectory flown in order to reconstruct the environment.

5.3.2 Research Outlook

In the previous section we showed that stable MAV navigation in large environments is already used to reconstruct a previously unknown environment in order to perform higher level tasks such as optimal coverage (Doitsidis et al. (2011)). Nevertheless, there are still fundamental questions to solve concerning both the sensor-fusion system as such and higher level tasks which require a robustly functioning MAV navigation in large environments.

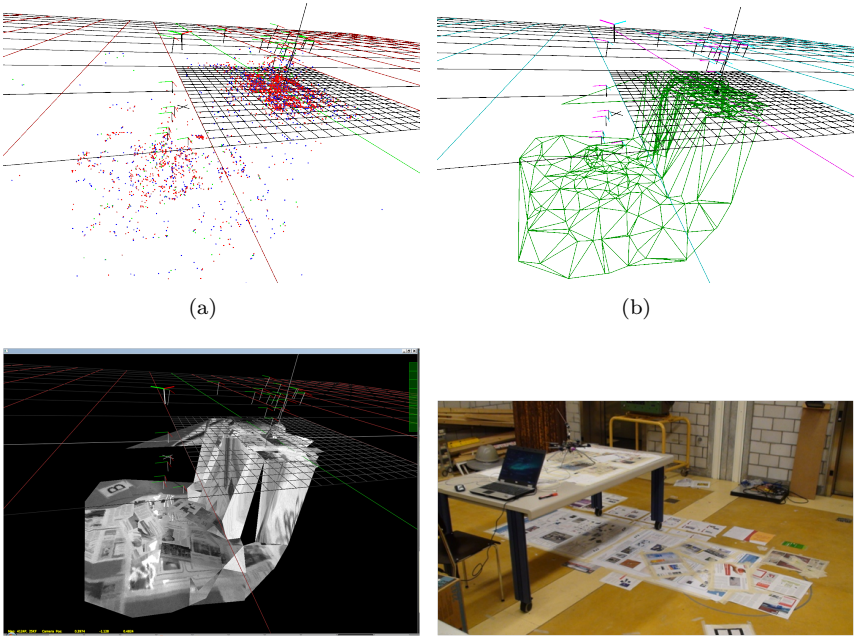


Figure 5.1: Representation of the different steps of meshing and texturing an environment with an airborne vehicle. The 3D point cloud in a) is obtained without additional computation power directly from the visual pose estimator used for vision based navigation in this dissertation. b) shows the meshed point cloud based on an real-time improved Delaunay meshing method. c) is the mesh textured aid of the key-frames stored in the visual pose estimator. d) picture of the original scene reconstructed in a), b), and c). (Weiss et al. (2011a))

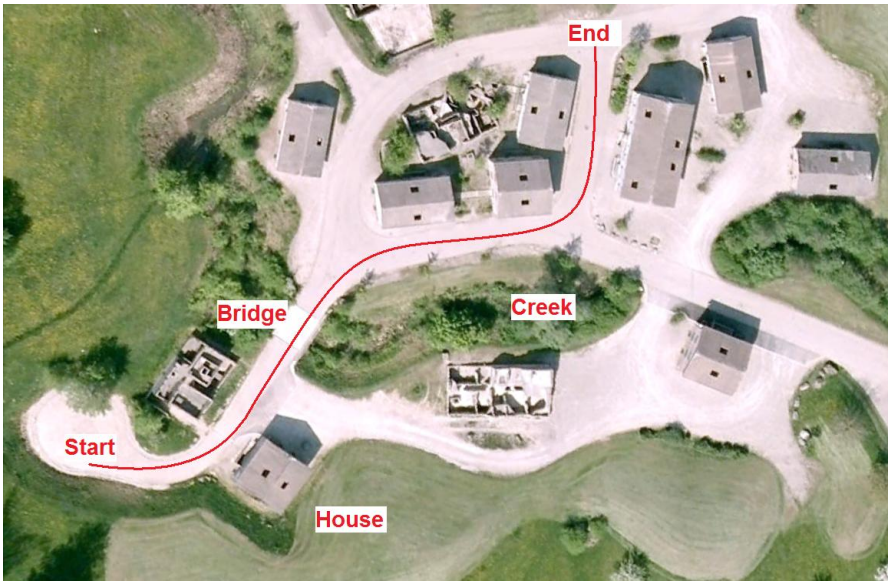
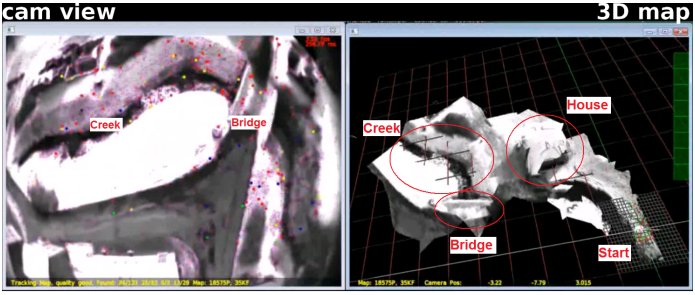
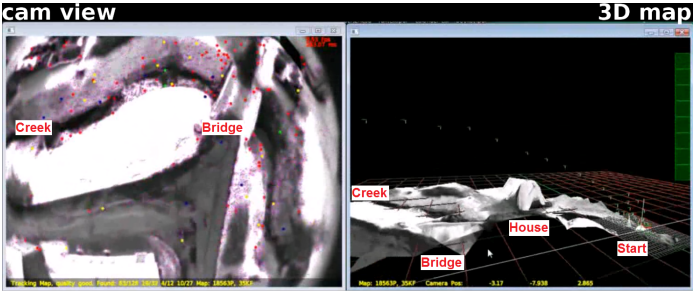


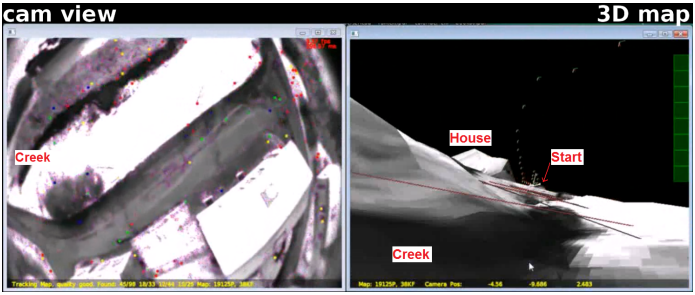
Figure 5.2: Outdoor flight path through the small village. Note the labeled positions for easier understanding of the following figures. The path length is over 100m and the flight height varies between 5 and 10m. (Google Maps)



(a)



(b)

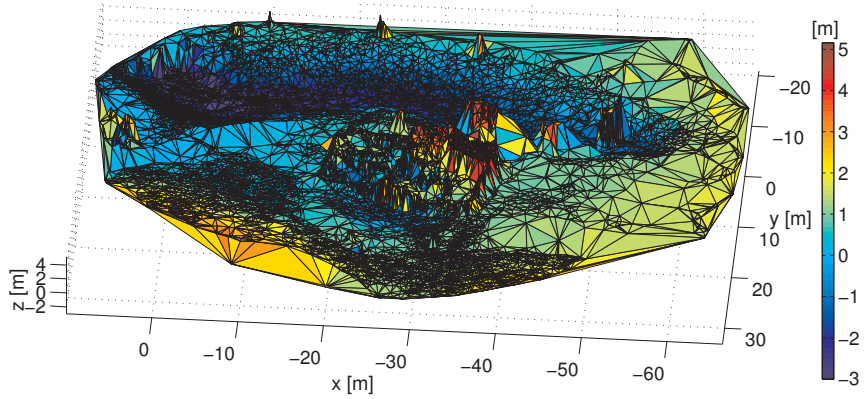


(c)

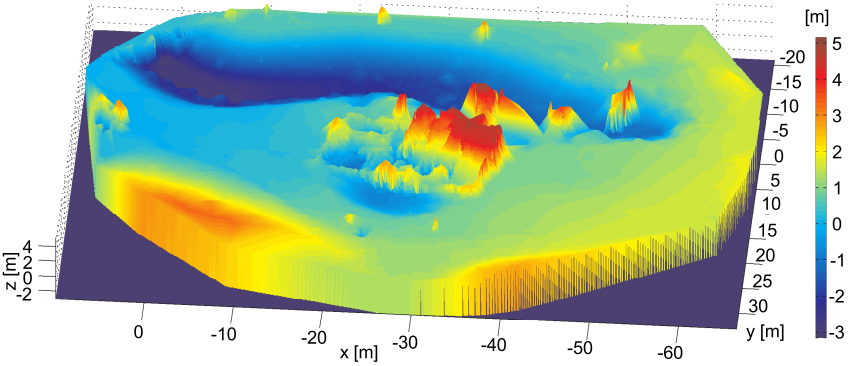
Figure 5.3: a) Overview of the first half of the total path. Compare the labeled locations with the ones in b) and c) b) Side view of the terrain to show the modeled house elevation as an obstacle to the MAV. Note that the elevation map is more accurate the more salient points the vision pose estimator detected on the object. The snow covered roof did not provide many such points and the model suffers on quality accordingly. Conversely, the creek in c) provides many features and the model is much more accurate. (Weiss et al. (2011a))

(a)

Figure 5.4: a) Overview of the reconstructed area. In red, the path flown by the MAV. (Google Maps)



(a)



(b)

Figure 5.5: a) reconstructed elevation mesh map of the area in Fig. 5.4 using the approach in (Weiss et al. (2011a)). b) interpolated dense elevation map for obstacle avoidance and optimal coverage tasks such as the one presented in (Doitsidis et al. (2011)).

On the sensor-fusion side, in this dissertation, we analyzed the theoretical track of analyzing the non-linear and time-continuous system. Furthermore, we tested the implemented system on the real platform under real conditions. However, we suggest to investigate further in the theoretical analysis of the linearized and discretized sensor-fusion system. We gave a short insight to the questions arising in this topic in Section 3.4.8. The most advanced work in this direction may be (Mourikis (June 2008)) and presents an excellent starting point for the full coverage of this topic. In particular, we see the need of solving the following questions on the linearized and discretized system:

- Optimal sensor frequency: It is evident, that, in our vision based case augmented with a GPS sensor, the GPS only needs to compensate for the very slow position and yaw drift of the vision system. Furthermore, we can assume that the faster the sampling rate the better the system behavior (as it approximates better the continuous case). The exact knowledge about the minimal (or optimal) sensor rate needed for a certain system behavior would help in path planning (e.g. GPS availability), in power saving (decrease sensor measurement rate) and in keeping the system stable. The work in (Mourikis (June 2008)) already tackles this issue to a large extent.

- Maximal noise value: certainly closely related to the optimal sensor frequency is the influence of noise on the state estimate. Knowledge on the maximal allowed noise level per sensor would facilitate the choice of sensors on the real platform and reveal particularly sensitive states.

- Fast filter convergence: While the analysis on the non-linear time-continuous system already reveals the necessity of some excitation of the system it is important to know which movement (based on the sensor frequency and noise level) lead to fastest convergence of a given state. This question eventually leads to the optimal path problem to reach a goal point while at the same time keep a low entropy on the state random variables. While this is tackled for small systems in (Martinelli and Siegwart (2006)) the solution presented by the authors is inapplicable for more complex systems. We suggest to investigate in the direction of the work of (Brian J Julian (September 2011)) applying information theoretic aspects and find a local best choice for the next excitation.

The solution of the above questions together with the analysis we provide in this dissertation yields a full recipe to analyze and implement a versatile and modular sensor-fusion framework. Whereas our contribution provides

detailed insight which *type* of sensors are needed to achieve the desired estimation properties (i.e. observable states) the solution to the above questions would lead to the specific hardware requirements (frequency, noise level) of such sensors.

The number of higher level tasks which require a robustly functioning MAV navigation in large environments is vast. Essentially, our proposed method of vision based navigation for micro helicopters renders the MAV close to a ground vehicle: all agile parts of the system are taken care of by our proposed method. The system has now a "stop" function keeping the MAV on place and simple goal vectors obtained from regular path planners are now sufficient to reach the desired location. With these abilities, we can port a vast amount of approaches from ground vehicles to aerial vehicles and extend them to full 3D.

Appendix A

Quantitative Results on the Real Platform

In the following, detailed results from the experiments are shown. The values denote the RMS error between the filter estimate and ground truth, except for the second vector in the columns for p_w^i and q_w^i which denote the RMS error between setpoint and ground truth. (w/) and (w/o) denote experiments with and without attitude measurements respectively (i.e. a 6DoF pose measurement or a 3DoF position measurement).

A.1 Hovering

Table A.1: Measurement Noise @ 10 Hz vs. RMS Error

σ_p[m] σ_q[rad]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
0.001 0.001 w/ att.	$\begin{bmatrix}0.006\\0.003\\0.001\end{bmatrix}$ $\begin{bmatrix}0.017\\0.019\\0.005\end{bmatrix}$	$\begin{bmatrix}0.003\\0.010\\0.019\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.015\end{bmatrix}$	0.3	$\begin{bmatrix}0.052\\0.032\\0.010\end{bmatrix}$	$\begin{bmatrix}0.025\\0.019\\0.000\end{bmatrix}$
0.005 0.009 w/ att.	$\begin{bmatrix}0.011\\0.014\\0.003\end{bmatrix}$ $\begin{bmatrix}0.028\\0.032\\0.006\end{bmatrix}$	$\begin{bmatrix}0.006\\0.007\\0.010\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.016\end{bmatrix}$	0.4	$\begin{bmatrix}0.085\\0.024\\0.050\end{bmatrix}$	$\begin{bmatrix}0.026\\0.010\\0.000\end{bmatrix}$
0.010 0.018 w/ att.	$\begin{bmatrix}0.014\\0.008\\0.006\end{bmatrix}$ $\begin{bmatrix}0.038\\0.025\\0.008\end{bmatrix}$	$\begin{bmatrix}0.007\\0.009\\0.019\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.022\end{bmatrix}$	0.4	$\begin{bmatrix}0.054\\0.052\\0.029\end{bmatrix}$	$\begin{bmatrix}0.031\\0.018\\0.000\end{bmatrix}$
0.020 0.036 w/ att.	$\begin{bmatrix}0.017\\0.012\\0.010\end{bmatrix}$ $\begin{bmatrix}0.056\\0.036\\0.013\end{bmatrix}$	$\begin{bmatrix}0.006\\0.059\\0.034\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.033\end{bmatrix}$	0.4	$\begin{bmatrix}0.020\\0.049\\0.026\end{bmatrix}$	$\begin{bmatrix}0.023\\0.042\\0.000\end{bmatrix}$
0.050 0.036 w/ att.	$\begin{bmatrix}0.034\\0.025\\0.026\end{bmatrix}$ $\begin{bmatrix}0.104\\0.043\\0.033\end{bmatrix}$	$\begin{bmatrix}0.011\\0.016\\0.009\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.015\end{bmatrix}$	0.6	$\begin{bmatrix}0.016\\0.029\\0.082\end{bmatrix}$	$\begin{bmatrix}0.022\\0.010\\0.000\end{bmatrix}$
0.100 0.036 w/o att.	$\begin{bmatrix}0.055\\0.054\\0.039\end{bmatrix}$ $\begin{bmatrix}0.098\\0.099\\0.062\end{bmatrix}$	$\begin{bmatrix}0.021\\0.025\\0.230\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.229\end{bmatrix}$	0.8	$\begin{bmatrix}0.005\\0.054\\0.113\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$
0.200 0.036 w/o att.	$\begin{bmatrix}0.087\\0.098\\0.080\end{bmatrix}$ $\begin{bmatrix}0.150\\0.166\\0.096\end{bmatrix}$	$\begin{bmatrix}0.045\\0.016\\0.190\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.202\end{bmatrix}$	0.8	$\begin{bmatrix}0.035\\0.057\\0.058\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$

Table A.2: Measurement Delay @ 10 Hz vs. RMS Error

delay [ms]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
0	$\begin{bmatrix}0.005\\0.011\\0.002\end{bmatrix}$ $\begin{bmatrix}0.029\\0.023\\0.006\end{bmatrix}$	$\begin{bmatrix}0.012\\0.041\\0.008\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.015\end{bmatrix}$	0.3	$\begin{bmatrix}0.041\\0.010\\0.004\end{bmatrix}$	$\begin{bmatrix}0.034\\0.007\\0.000\end{bmatrix}$
50	$\begin{bmatrix}0.006\\0.012\\0.004\end{bmatrix}$ $\begin{bmatrix}0.035\\0.026\\0.008\end{bmatrix}$	$\begin{bmatrix}0.007\\0.020\\0.070\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.065\end{bmatrix}$	0.3	$\begin{bmatrix}0.028\\0.002\\0.064\end{bmatrix}$	$\begin{bmatrix}0.021\\0.071\\0.001\end{bmatrix}$
100	$\begin{bmatrix}0.011\\0.015\\0.005\end{bmatrix}$ $\begin{bmatrix}0.036\\0.036\\0.009\end{bmatrix}$	$\begin{bmatrix}0.012\\0.052\\0.025\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.024\end{bmatrix}$	0.3	$\begin{bmatrix}0.037\\0.005\\0.069\end{bmatrix}$	$\begin{bmatrix}0.015\\0.026\\0.000\end{bmatrix}$
200	$\begin{bmatrix}0.009\\0.016\\0.004\end{bmatrix}$ $\begin{bmatrix}0.036\\0.036\\0.006\end{bmatrix}$	$\begin{bmatrix}0.016\\0.077\\0.009\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.016\end{bmatrix}$	0.3	$\begin{bmatrix}0.031\\0.010\\0.076\end{bmatrix}$	$\begin{bmatrix}0.015\\0.006\\0.000\end{bmatrix}$
500	$\begin{bmatrix}0.017\\0.021\\0.005\end{bmatrix}$ $\begin{bmatrix}0.066\\0.080\\0.012\end{bmatrix}$	$\begin{bmatrix}0.010\\0.059\\0.040\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.045\end{bmatrix}$	0.4	$\begin{bmatrix}0.031\\0.002\\0.020\end{bmatrix}$	$\begin{bmatrix}0.040\\0.040\\0.001\end{bmatrix}$

Table A.3: Measurement Update Rate vs. RMS Error

update [ms]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
20	$\begin{bmatrix}0.004\\0.012\\0.001\end{bmatrix}$ $\begin{bmatrix}0.026\\0.018\\0.005\end{bmatrix}$	$\begin{bmatrix}0.012\\0.065\\0.026\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.032\end{bmatrix}$	0.3	$\begin{bmatrix}0.041\\0.009\\0.003\end{bmatrix}$	$\begin{bmatrix}0.034\\0.025\\0.000\end{bmatrix}$
10	$\begin{bmatrix}0.005\\0.011\\0.001\end{bmatrix}$ $\begin{bmatrix}0.029\\0.023\\0.006\end{bmatrix}$	$\begin{bmatrix}0.012\\0.041\\0.008\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.014\end{bmatrix}$	0.3	$\begin{bmatrix}0.041\\0.010\\0.005\end{bmatrix}$	$\begin{bmatrix}0.034\\0.007\\0.000\end{bmatrix}$
5	$\begin{bmatrix}0.005\\0.012\\0.002\end{bmatrix}$ $\begin{bmatrix}0.021\\0.032\\0.005\end{bmatrix}$	$\begin{bmatrix}0.012\\0.043\\0.018\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.016\end{bmatrix}$	0.3	$\begin{bmatrix}0.039\\0.009\\0.009\end{bmatrix}$	$\begin{bmatrix}0.040\\0.017\\0.000\end{bmatrix}$
2	$\begin{bmatrix}0.009\\0.014\\0.003\end{bmatrix}$ $\begin{bmatrix}0.032\\0.030\\0.008\end{bmatrix}$	$\begin{bmatrix}0.013\\0.046\\0.022\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.021\end{bmatrix}$	0.3	$\begin{bmatrix}0.039\\0.008\\0.013\end{bmatrix}$	$\begin{bmatrix}0.041\\0.023\\0.000\end{bmatrix}$
1	$\begin{bmatrix}0.019\\0.016\\0.005\end{bmatrix}$ $\begin{bmatrix}0.045\\0.039\\0.009\end{bmatrix}$	$\begin{bmatrix}0.012\\0.019\\0.005\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.017\end{bmatrix}$	0.3	$\begin{bmatrix}0.040\\0.016\\0.024\end{bmatrix}$	$\begin{bmatrix}0.029\\0.003\\0.000\end{bmatrix}$

A.2 Dynamic Flight

Table A.4: Measurement Noise @ 10 Hz vs. RMS Error

σ_p[m] σ_q[rad]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
.001 .001 w/o att.	$\begin{bmatrix}0.003\\0.002\\0.001\end{bmatrix}$ $\begin{bmatrix}0.049\\0.055\\0.007\end{bmatrix}$	$\begin{bmatrix}0.024\\0.027\\0.040\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.033\end{bmatrix}$	0.4	$\begin{bmatrix}0.024\\0.016\\0.015\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$
.005 .009 w/ att.	$\begin{bmatrix}0.014\\0.011\\0.005\end{bmatrix}$ $\begin{bmatrix}0.059\\0.076\\0.015\end{bmatrix}$	$\begin{bmatrix}0.014\\0.026\\0.015\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.028\end{bmatrix}$	1.1	$\begin{bmatrix}0.041\\0.015\\0.017\end{bmatrix}$	$\begin{bmatrix}0.022\\0.019\\0.000\end{bmatrix}$
.005 .009 w/o att.	$\begin{bmatrix}0.014\\0.009\\0.005\end{bmatrix}$ $\begin{bmatrix}0.082\\0.086\\0.015\end{bmatrix}$	$\begin{bmatrix}0.022\\0.029\\0.029\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.034\end{bmatrix}$	0.5	$\begin{bmatrix}0.016\\0.021\\0.037\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$
.010 .018 w/ att.	$\begin{bmatrix}0.016\\0.010\\0.005\end{bmatrix}$ $\begin{bmatrix}0.070\\0.036\\0.011\end{bmatrix}$	$\begin{bmatrix}0.016\\0.024\\0.015\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.026\end{bmatrix}$	1.1	$\begin{bmatrix}0.037\\0.017\\0.025\end{bmatrix}$	$\begin{bmatrix}0.018\\0.006\\0.000\end{bmatrix}$
.010 .018 w/o att.	$\begin{bmatrix}0.016\\0.012\\0.007\end{bmatrix}$ $\begin{bmatrix}0.071\\0.056\\0.017\end{bmatrix}$	$\begin{bmatrix}0.021\\0.025\\0.049\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.039\end{bmatrix}$	0.6	$\begin{bmatrix}0.017\\0.028\\0.067\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$
.020 .036 w/ att.	$\begin{bmatrix}0.017\\0.017\\0.012\end{bmatrix}$ $\begin{bmatrix}0.061\\0.052\\0.020\end{bmatrix}$	$\begin{bmatrix}0.015\\0.028\\0.025\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.039\end{bmatrix}$	1.2	$\begin{bmatrix}0.031\\0.034\\0.025\end{bmatrix}$	$\begin{bmatrix}0.016\\0.012\\0.000\end{bmatrix}$
.020 .036 w/o att.	$\begin{bmatrix}0.018\\0.014\\0.011\end{bmatrix}$ $\begin{bmatrix}0.072\\0.075\\0.016\end{bmatrix}$	$\begin{bmatrix}0.021\\0.028\\0.038\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.037\end{bmatrix}$	0.7	$\begin{bmatrix}0.016\\0.040\\0.062\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$
.050 .036 w/ att.	$\begin{bmatrix}0.033\\0.033\\0.020\end{bmatrix}$ $\begin{bmatrix}0.105\\0.088\\0.026\end{bmatrix}$	$\begin{bmatrix}0.014\\0.027\\0.018\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.026\end{bmatrix}$	1.2	$\begin{bmatrix}0.018\\0.044\\0.026\end{bmatrix}$	$\begin{bmatrix}0.016\\0.014\\0.000\end{bmatrix}$
.050 .036 w/o att.	$\begin{bmatrix}0.033\\0.035\\0.020\end{bmatrix}$ $\begin{bmatrix}0.098\\0.108\\0.024\end{bmatrix}$	$\begin{bmatrix}0.023\\0.029\\0.060\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.050\end{bmatrix}$	0.7	$\begin{bmatrix}0.018\\0.024\\0.062\end{bmatrix}$	$\begin{bmatrix}0.000\\0.000\\0.000\end{bmatrix}$

Table A.5: Measurement Delay @ 10 Hz vs. RMS Error

delay [ms]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
50	$\begin{bmatrix}0.009\\0.007\\0.004\end{bmatrix}$ $\begin{bmatrix}0.060\\0.080\\0.007\end{bmatrix}$	$\begin{bmatrix}0.021\\0.029\\0.050\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.045\end{bmatrix}$	0.500	$\begin{bmatrix}0.010\\0.043\\0.018\end{bmatrix}$	$\begin{bmatrix}0.025\\0.032\\0.000\end{bmatrix}$
100	$\begin{bmatrix}0.007\\0.009\\0.004\end{bmatrix}$ $\begin{bmatrix}0.064\\0.092\\0.006\end{bmatrix}$	$\begin{bmatrix}0.025\\0.044\\0.014\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.034\end{bmatrix}$	0.700	$\begin{bmatrix}0.006\\0.008\\0.052\end{bmatrix}$	$\begin{bmatrix}0.021\\0.024\\0.000\end{bmatrix}$
200	$\begin{bmatrix}0.025\\0.027\\0.006\end{bmatrix}$ $\begin{bmatrix}0.149\\0.068\\0.013\end{bmatrix}$	$\begin{bmatrix}0.030\\0.041\\0.036\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.038\end{bmatrix}$	1.000	$\begin{bmatrix}0.016\\0.014\\0.031\end{bmatrix}$	$\begin{bmatrix}0.010\\0.039\\0.000\end{bmatrix}$
500	$\begin{bmatrix}0.095\\0.170\\0.030\end{bmatrix}$ $\begin{bmatrix}0.264\\0.400\\0.036\end{bmatrix}$	$\begin{bmatrix}0.024\\0.063\\0.050\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.066\end{bmatrix}$	1.100	$\begin{bmatrix}0.018\\0.021\\0.030\end{bmatrix}$	$\begin{bmatrix}0.011\\0.046\\0.000\end{bmatrix}$

Table A.6: Update rate vs. RMS Error

update [ms]	p_w^i[m]	q_w^i[rad (rpy)]	λ[%]	p_i^s[m]	q_i^s[rad]
10	$\begin{bmatrix}0.009\\0.006\\0.005\end{bmatrix}$ $\begin{bmatrix}0.078\\0.064\\0.012\end{bmatrix}$	$\begin{bmatrix}0.020\\0.032\\0.020\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.025\end{bmatrix}$	0.8	$\begin{bmatrix}0.011\\0.009\\0.013\end{bmatrix}$	$\begin{bmatrix}0.019\\0.011\\0.000\end{bmatrix}$
5	$\begin{bmatrix}0.007\\0.006\\0.004\end{bmatrix}$ $\begin{bmatrix}0.050\\0.060\\0.008\end{bmatrix}$	$\begin{bmatrix}0.023\\0.040\\0.042\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.047\end{bmatrix}$	0.8	$\begin{bmatrix}0.007\\0.019\\0.024\end{bmatrix}$	$\begin{bmatrix}0.022\\0.024\\0.000\end{bmatrix}$
2	$\begin{bmatrix}0.016\\0.015\\0.005\end{bmatrix}$ $\begin{bmatrix}0.094\\0.068\\0.015\end{bmatrix}$	$\begin{bmatrix}0.012\\0.026\\0.031\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.031\end{bmatrix}$	0.8	$\begin{bmatrix}0.030\\0.014\\0.049\end{bmatrix}$	$\begin{bmatrix}0.013\\0.007\\0.000\end{bmatrix}$
1	$\begin{bmatrix}0.050\\0.032\\0.009\end{bmatrix}$ $\begin{bmatrix}0.177\\0.100\\0.020\end{bmatrix}$	$\begin{bmatrix}0.026\\0.040\\0.059\end{bmatrix}$ $\begin{bmatrix}-\\-\\0.063\end{bmatrix}$	0.8	$\begin{bmatrix}0.014\\0.029\\0.123\end{bmatrix}$	$\begin{bmatrix}0.016\\0.024\\0.000\end{bmatrix}$

Bibliography

N. Abdelkrim, N. Aouf, A. Tsourdos, and B. White. Robust nonlinear filtering for ins/gps uav localization. In *Mediterranean Conference on Control and Automation*, pages 695–702, June 2008.

M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo Vision and Laser Odometry for Autonomous Helicopters in GPS-denied Indoor Environments. In *Proceedings of the SPIE Unmanned Systems Technology XI*, 2009.

M. C. Achtelik, J. Stumpf, D. Gurdan, and K.-M. Doth. Design of a flexible high performance quadcopter platform breaking the mav endurance record with laser power beaming. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5166 –5172, sept. 2011a. doi: 10.1109/IROS.2011.6094731.

M. W. Achtelik, M. C. Achtelik, S. Weiss, and R. Siegwart. Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011b.

S. Ahrens, D. Levine, G. Andrews, and J. How. Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments. In *International Conference on Robotics and Automation*, May 2009.

E. Altug, J. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *International Conference on Robotics and Automation*, pages 72–77, May 2002.

L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. In *Proceedings of The IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004.

L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *Int. J. Rob. Res.*, 26(6):577–589, 2007. ISSN 0278-3649. doi: http://dx.doi.org/10.1177/0278364907079283.

J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragon, C. Martinez, and M. Olivares. Visual 3-d slam from uavs. *Journal of Intelligent and Robotic Systems*, 2008.

Ascending Technologies GmbH. website. http://www.asctec.de.

A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructed and unknown indoor environments. In *European Conference on Micro Aerial Vechicles (EMAV)*, 2009a.

A. Bachrach, R. He, and N. Roy. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 2009b.

G. Baldwin, R. Mahony, and J. Trumpf. A nonlinear observer for 6 DOF pose estimation from inertial and bearing measurements. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Kobe, 2009.

H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008.

C. Beder and R. Steffen. *Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence.* Number 4174 in LNCS. Springer, 2006.

A. Beyeler. *Vision-based control of near-obstacle flight.* PhD thesis, Lausanne, 2009. URL http://library.epfl.ch/theses/?nr=4456,http://library.epfl.ch/theses/?nr=4456.

G. Bleser and G. Hendeby. Using optical flow for filling the gaps in visual-inertial tracking. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2010.

M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 21–28, 2010.

S. Bouabdallah. *Design and control of quadrotors with application to autonomous flying*. PhD thesis, Lausanne, 2007. URL http://library.epfl.ch/theses/?nr=3727,http://library.epfl.ch/theses/?nr=3727.

S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *International Conference on Robotics and Automation*, pages 2247–2252, Apr. 2005.

S. Bouabdallah and R. Siegwart. Design and control of a miniature quadrotor. 33:171–210, 2007.

B. G. Breitmeyer and L. Ganz. Temporal studies with flashed gratings: Inferences about human transient and sustained channels. *Vision Research*, 17 (7):861 – 865, 1977. ISSN 0042-6989. doi: 10.1016/0042-6989(77)90130-4. URL http://www.sciencedirect.com/science/article/pii/0042698977901304.

M. S. D. R. Brian J Julian, Michael Angermann. A scalable information theoretic approach to distributed robot coordination. In *IEEE International Conference on Intelligent Robot And Systems (IROS)*, September 2011.

D. A. M. Bronstein. Vertigo. its multisensory syndromes. *Brain*, 123(4):849–851, 2000. doi: 10.1093/brain/123.4.849-a. URL http://brain.oxfordjournals.org/content/123/4/849.2.short.

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. pages 3143–3148, 2009. URL http://portal.acm.org/citation.cfm?id=1703775.1703957.

G. Cai, A. Cai, B. Chen, and T. Lee. Construction, modeling and control of a mini autonomous uav helicopter. In *International Conference on Automation and Logistics*, pages 449–454, Sept. 2008.

F. W. Campbell and R. H. Wurtz. Saccadic omission: Why we do not see a grey-out during a saccadic eye movement. *Vision Research*, 18(10):1297 – 1303, 1978. ISSN 0042-6989. doi: 10.1016/0042-6989(78)90219-5. URL http://www.sciencedirect.com/science/article/pii/0042698978902195.

P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini-rotorcraft having four rotors. In *International Conference on Intelligent Robots and Systems*, pages 2693–2698, Sept. 2004.

J. Chen and D. Dawson. Uav tracking with a monocular camera. In *Conference on Decision and Control*, 2006.

T. Cheviron, T. Hamel, R. Mahony, and G. Baldwin. Robust nonlinear fusion of inertial and visual data for position, velocity and attitude estimation of uav. In *International Conference on Robotics and Automation*, pages 2010–2016, Apr. 2007.

S. Chroust and M. Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotic Systems*, 21(2):73–83, 2004.

J. Civera, O. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point RANSAC for EKF filtering. application to real-time structure from motion and visual odometry. Accepted for publication in the Journal of Field Robotics, 2010.

P. Corke. An inertial and visual sensing system for a small autonomous helicopter. *International Journal of Robotics Systems*, 21(2):43–51, 2004.

P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *International Journal of Robotics Research*, 26(6):519–535, 2007.

A. J. Davison. *Mobile Robot Navigation Using Active Vision.* PhD thesis, University of Oxford, 1998.

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4.

A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007.

L. Doitsidis, A. Renzaglia, S. Weiss, E. Kosmatopoulos, D. Scaramuzza, and R. Siegwart. 3d surveillance coverage using maps extracted by a monocular slam algorithm. In *International Conference on Robotics and Intelligent System (IROS)*, San Francisco, September 2011.

A. Eudes and M. Lhuillier. *Error propagations for local bundle adjustment*, volume 0. IEEE Computer Society, Los Alamitos, CA, USA, 2009. ISBN 978-1-4244-3992-8. doi: http://doi.ieeecomputersociety.org/10.1109/CVPRW.2009.5206824.

S. Fowers, D. Lee, B. Tippets, K. Lillywhite, A. Dennis, and J. Archibald. Vision aided stabilization and the development of a quad-rotor micro uav. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 143–148, June 2007.

P. Furgale and T. D. Barfoot. Visual teach and repeat for long-range rover autonomy. volume 27, pages 534–560, Chichester, UK, September 2010. John Wiley and Sons Ltd. doi: http://dx.doi.org/10.1002/rob.v27:5. URL http://dx.doi.org/10.1002/rob.v27:5.

M. Garratt and J. Chahl. Vision-based terrain following for an unmanned rotorcraft. *Journal of Field Robotics*, 25(4-5):284–301, 2008.

P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous motion and structure estimation by fusion of inertial and vision data. *The International Journal of Robotics Research*, 26(6):591–605, 2007.

T. Hamel, R. Mahony, and A. Chriette. Visual servo trajectory tracking for a four rotor vtol aerial vehicle. In *International Conference on Robotics and Automation*, pages 2781–2786, May 2002.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conference*, pages 147–âĂŞ151, 1988.

B. Herisse, F. Russotto, T. Hamel, and R. Mahony. Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow. In *International Conference on Intelligent Robots and Systems*, pages 801–806, Sept. 2008.

B. Hérissé, T. Hamel, R. Mahony, and F.-X. Russotto. A terrain-following control approach for a VTOL unmanned aerial vehicle using average optical flow. *Autonomous Robots*, 29:381–399, 2010.

R. Hermann and A. Krener. Nonlinear controllability and observability. *Automatic Control, IEEE Transactions on*, 22(5):728 – 740, Oct. 1977. ISSN 0018-9286. doi: 10.1109/TAC.1977.1101601.

J. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems Magazine*, 2008.

S. Hrabar, G. Sukhatme, P. corke, K. Usher, and J. Roberts. Combined optic-flow and stereo-based navigation of urban canyons for a uav. In *International Conference on Intelligent Robots and Systems*, pages 3309–3316, Aug. 2005.

A. Huster, E. W. Frew, and S. M. Rock. Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements. In *Proceedings of The Oceans Conference*, volume 3, pages 1857–1864, Biloxi, 2002. MTS/IEEE.

J. J. L. Center and K. H. Knuth. Bayesian visual odometry. *American Institude of Physics (AIP) Conference Proceedings*, 1305(1):75–82, 2011.

E. Jones. *Large Scale Visual Navigation and Community Map Building*. PhD thesis, University of California at Los Angeles, June 2009.

E. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *International Journal of Robotics Research*, 2010.

J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research (IJRR)*, 30(1):56–79, 2011.

G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

G. Klein and D. W. Murray. Parallel tracking and mapping on a camera phone. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009.

S. Klose, J. W. M., A. G., P. F. Holzapfel, and A. Knoll. Markerless, vision-assisted flight control of a quadrocopter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart, Y. Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence. In *International Conference on Robotics and Automation*, Shanghai, China, May 2011. URL http://hal.inria.fr/hal-00641772/en/.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

B. Ludington, E. Johnson, and G. Vachtsevanos. Augmenting uav autonomy. 24(5):63–71, Sept. 2006.

S. Lupashin, A. Schoellig, M. Sherback, and R. D'Andrea. A simple learning strategy for high-speed quadrocopter multi-flips. In *International Conference on Robotics and Automation*, 2010.

T. Lupton and S. Sukkarieh. Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In *International Conference on Robotics and Automation*, Pasadena, California, USA, 2008.

T. Lupton and S. Sukkarieh. Efficient integration of inertial observations into visual SLAM without initialization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009.

S. Lynen. Improving ptam to allow operation in large scale environments. April 2011.

Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3D vision : from images to geometric models*. Springer, 2000.

E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1843 – 1848 Vol.2, 26-may 1, 2004. doi: 10.1109/ROBOT.2004.1308092.

A. Martinelli. State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *Robotics, IEEE Transactions on*, 27(2):239 –255, april 2011. ISSN 1552-3098. doi: 10.1109/TRO.2011.2109210.

A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale and bias determination. *Transaction on Robotics*, 28 (Issue 1):44–60, February 2012.

A. Martinelli and R. Siegwart. Observability properties and optimal trajectories for on-line odometry self-calibration. In *Decision and Control, 2006 45th IEEE Conference on*, pages 3065 –3070, dec. 2006. doi: 10.1109/CDC.2006.377161.

A. Martinelli, C. Troiani, and A. Renzaglia. Vision-aided inertial navigation: Closed-form determination of absolute scale, speed and attitude. In *International Conference on Intelligent Robots and Systems*, San Francisco, États-Unis, 2011. URL http://hal.inria.fr/hal-00641050.

P. S. Maybeck. *Stochastic models, estimation, and control*, volume 1. Academic Press, 1979.

L. Mejias, K. Usher, J. Roberts, and P. Corke. Two seconds to touchdown – vision-based controlled forced landing. In *International Conference on Intelligent Robots and Systems*, pages 3527–3532, Oct. 2006.

D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520 –2525, may 2011. doi: 10.1109/ICRA.2011.5980409.

D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In *International Symposium on Experimental Robotics (ISER)*, 2010a.

D. Mellinger, M. Shomin, N. Michael, and V. Kumar. Cooperative grasping and transport using multiple quadrotors. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2010b.

N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 2010a.

N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The grasp multiple micro uav testbed. *IEEE Robotics and Automation Magazine*, May 2010b.

F. Mirzaei and S. Roumeliotis. 1|a kalman filter-based algorithm for imu-camera calibration. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2427 –2434, 292007-nov.2 2007. doi: 10.1109/IROS.2007.4399342.

F. Mirzaei and S. Roumeliotis. A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. *IEEE Transactions on Robotics and Automation*, 24(5):1143 –1156, 2008.

A. Mourikis. *Characterization and Optimization of the Accuracy of Mobile Robot Localization.* PhD thesis, Department of Computer Science and Engineering, University of Minnesota, June 2008.

A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, Roma, 2007.

A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam : Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, volume 1, 2011. URL http://www.doc.ic.ac.uk/~ajd/Publications/newcombe_etal_iccv2011.pdf.

G. Nutzi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of Intelligent and Robotic Systems*, November 2010.

A. Oosedo, A. Konno, T. Matumoto, K. Go, K. Masuko, S. Abiko, and M. Uchiyama. Design and simulation of a quad rotor tail-sitter unmanned aerial vehicle. In *System Integration (SII), 2010 IEEE/SICE International Symposium on*, pages 254 –259, dec. 2010. doi: 10.1109/SII.2010.5708334.

S. Park, D. Won, M. Kang, T. Kim, H. Lee, and S. Kwon. Ric (robust internal-loop compensator) based flight control of a quad-rotor type uav. In *International Conference on Intelligent Robots and Systems*, pages 3542–3547, Aug. 2005.

A. R. Parker. On the origin of optics. volume 43, pages 323 – 329. 2011. doi: 10.1016/j.optlastec.2008.12.020. URL http://www.sciencedirect.com/science/article/pii/S0030399208002454. <ce:title>Colour and Design II: Colour in plants and animals - Inspiration for Design</ce:title>.

K. Peng, M. Dong, B. Chen, G. Cai, Y. Lum, and T. Lee. Design and implementation of a fully autonomous flight control system for a uav helicopter. In *Chinese Control Conference*, pages 662–667, July 2007.

P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardï£¡s. Inertial aiding of inverse depth slam using a monocular camera. In *ICRA*, pages 2797–2802. IEEE, 2007.

A. Proctor and E. Johnson. Vision-only aircraft flight control methods and test results. In *AIAA Guidance, Navigation, Control Conference*, 2004.

G. Qian, R. Chellappa, and Q. Zheng. Bayesian structure from motion using inertial information. In *International Conference on Image Processing*, Rochester, New York, USA, 2002.

O. A. Rawashdeh, H. C. Yang, R. D. AbouSleiman, and B. H. Sababha. Microraptor: A low-cost autonomous quadrotor system. *ASME Conference Proceedings*, 2009(49002):567–574, 2009. doi: 10.1115/DETC2009-86490. URL http://link.aip.org/link/abstract/ASMECP/v2009/i49002/p567/s1.

K. J. Röbenack, Klaus; Reinschke. The computation of lie derivatives and lie brackets based on automatic differentiation. *Journal of Applied Mathematics and Mechanics (ZAMM)*, 2:114–123, 2004.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Proceedings of The IEEE International Conference on Robotics and Automation*, pages 4326–4333, Washington D.C., 2002.

F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In *International Conference on Robotics and Automation*, pages 2339–2346, Apr. 2004.

S. Saripalli, J. Montgomery, and G. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *International Conference on Robotics and Automation*, pages 2799–2804, May 2002.

D. Schafroth. *Aerodynamics, modeling and control of an autonomous micro helicopter*. PhD thesis, 2010.

F. Schill, R. Mahony, and P. Corke. Estimating ego-motion in panoramic image sequences with inertial measurements. In *Robotics Research*, volume 70, pages 87–101. Springer Berlin / Heidelberg, 2011.

J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.

G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *In Proceedings of Robotics: Science and Systems*, pages 976–982, 2009.

J. A. Simmons and R. A. Stein. Acoustic imaging in bat sonar: Echolocation signals and the evolution of echolocation. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 135: 61–84, 1980. ISSN 0340-7594. URL http://dx.doi.org/10.1007/BF00660182. 10.1007/BF00660182.

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular slam: Why filter? In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

D. Strelow. *Motion estimation from image and inertial measurements*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, November 2004.

D. Strelow and S. Singh. Optimal motion estimation from visual and inertial measurements. In *Proceedings of the Sixth IEEE Workshop on Applications*

of Computer Vision, WACV '02, pages 314–, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1858-3. URL http://portal.acm.org/citation.cfm?id=832302.836841.

D. Strelow and S. Singh. Online motion estimation from image and inertial measurements. In *Workshop on Integration of Vision and Inertial Sensors (INERVIS)*, Coimbra, Portugal, 2003.

J. Sturm and A. Visser. An appearance-based visual compass for mobile robots. volume 57, pages 536 – 545, 2009. doi: DOI: 10.1016/j.robot.2008.10.002. URL http://www.sciencedirect.com/science/article/B6V16-4TSTY9B-1/2/90b2ad61d3e7076f9f108680284e68fe.

T. Templeton, D. Shim, C. Geyer, and S. Sastry. Autonomous vision-based landing and terrain mapping using an mpc-controlled unmanned rotorcraft. In *International Conference on Robotics and Automation*, pages 1349–1356, Apr. 2007.

N. Trawny and S. Roumeliotis. Indirect kalman filter for 3d attitude estimation. Technical report, University of Minnesota, Department of Computing Science and Engineering, 2005.

M. Valenti, B. Bethke, G. Fiore, , and J. P. How. Indoor multivehicle flight testbed for fault detection, isolation and recovery. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, May 2006.

S. Weiss and R. Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart. Intuitive 3d maps for mav terrain exploration and obstacle avoidance. *Journal of Intelligent & Robotic Systems*, 61:473–493, 2011a. ISSN 0921-0296. URL http://dx.doi.org/10.1007/s10846-010-9491-y. 10.1007/s10846-010-9491-y.

S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011b. ISSN 1556-4967. doi: 10.1002/rob.20412. URL http://dx.doi.org/10.1002/rob.20412.

S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart. Versatile distributed pose estimation and sensor self-calibration for an autonomous mav. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012a.

S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012b.

K. E. Wenzel, A. Masselli, , and A. Zell. Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle. In *UAV'10 3rd International Symposium on Unmanned Aerial Vehicles*, 2010.

W. Wiltschko and R. Wiltschko. Magnetic orientation in birds. *Journal of Experimental Biology*, 199(1):29–38, 1996. URL http://jeb.biologists.org/content/199/1/29.abstract.

B. Yun, K. Peng, and B. Chen. Enhancement of gps signals for automatic control of a uav helicopter system. In *International Conference on Robotics and Automation*, pages 1185–1189, June 2007.

S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. Mav navigation through indoor corridors using optical flow. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3361–3368, 2010.

J. Zufferey. *Bio-inspired vision-based flying robots*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2005.

J. Zufferey and D. Floreano. Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

J. Zufferey and D. Floreano. Fly-inspired visual steering of an ultralight indoor aircraft. 22(1):137–146, 2006.

J.-C. Zufferey, A. Klaptocz, A. Beyeler, J.-D. Nicoud, and D. Floreano. A 10-gram Vision-based Flying Robot. *Advanced Robotics*, 21(14):1671–1684, 2007. doi: 10.1163/156855307782227417. URL http://www.springerlink.com/content/9153v4j17j58289p/. Special issue on IROS2006.

J.-C. Zufferey, A. Beyeler, and D. Floreano. Autonomous flight at low altitude using light sensors and little computational power. *International Journal of Micro Air Vehicles*, 2(2):107–117, 2010. doi: 10.1260/1756-8293.2.2.107. URL http://lis.epfl.ch/microflyers.

Curriculum Vitae

Stephan M. Weiss was born in Caracas, Venezuela on July 16th, 1981. He has a Master of Science in Electrical Engineering and Information Technology from the Eidgenössische Technische Hochschule (ETH) Zurich. He is currently a PhD candidate at the Autonomous Systems Lab at ETH Zurich under Prof. Dr. Roland Siegwart.

The research interests of Stephan Weiss are in the area of autonomous navigation for MAVs. In particular he is interested in the state estimation of airborne vehicles in order to navigate them over long time in large and unknown areas. He is convinced that – taking nature as example – all basic maneuvers can be performed while only using one camera and an IMU as core sensors.

Previous projects include a pan-tilt system for an optical flow sensor which was used for ground robot navigation at the Institute of Neuroscience (INI) at ETH/University of Zurich and an embedded object segmentation and tracking algorithm using a range camera at ABB Switzerland. From August to November 2011 he visited the group of Prof. Dr. Daniela Rus, Distributed Robotics Lab (DRL) at MIT, Boston where he worked with Brian Julian and Andreas Breitenmoser on a probabilistic and distributed method for optimal coverage for MAVs. In July 2011 he gave a talk at the sFly summer school at ETH Zurich and in autumn 2010 and 2011 he gave the part on state estimation of the Masters course lecture "Unmanned Aerial Vehicles Design, Modeling and Control" at ETH Zurich.

Stephan Weiss is a fellow of UNITECH International and received the Oehler Award for exceptional achievents in robotics during highschool.

List of Publications

Journals

- Doitsidis, L.; Weiss, S.; Renzaglia, A.; Achtelik, W. M.; Kosmatopoulos, E.; Siegwart, R. & Scaramuzza, D. "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision" *Autonomous Robots, Springer Netherlands*, 2012, online first
- Weiss, S.; Scaramuzza, D. & Siegwart, R. "Monocular-SLAM based navigation for autonomous micro helicopters in GPS-denied environments" *Journal of Field Robotics, Wiley Subscription Services, Inc., A Wiley Company*, 2011, 28, 854-874
- Weiss, S.; Achtelik, M.; Kneip, L.; Scaramuzza, D. & Siegwart, R. "Intuitive 3D Maps for MAV Terrain Exploration and Obstacle Avoidance" *Journal of Intelligent & Robotic Systems, Springer Netherlands*, 2011, 61, 473-493
- Eberli, D.; Scaramuzza, D.; Weiss, S. & Siegwart, R. "Vision Based Position Control for MAVs Using One Single Circular Landmark" *Journal of Intelligent & Robotic Systems, Springer Netherlands*, 2011, 61, 495-512
- Nützi, G.; Weiss, S.; Scaramuzza, D. & Siegwart, R. "Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM" *Journal of Intelligent & Robotic Systems, Springer Netherlands*, 2011, 61, 287-299

Conferences

- Weiss, S.; Achtelik, M. W.; Lynen, S.; Chli, M. & Siegwart, R. "Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments" *IEEE International Conference on Robotics and Automation (ICRA)*, 2012 *Finalist Best Robotic Vision Paper*
- Weiss, S.; Achtelik, M. W.; Chli, M. & Siegwart, R. "Versatile Distributed Pose Estimation and Sensor Self-Calibration for Autonomous MAVs" *IEEE International Conference on Robotics and Automation (ICRA)*, 2012
- Achtelik, M.; Achtelik, M.; Weiss, S. & Siegwart, R. "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments" *IEEE International Conference on Robotics and Automation (ICRA)*, 2011
- Achtelik, M.; Weiss, S.; Chli, M.; Dellaert, F. & Siegwart, R. "Collaborative Stereo" *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011
- Doitsidis, L.; Renzaglia, A.; Weiss, S.; Kosmatopoulos, E.; Scaramuzza, D. & Siegwart, R. "3D Surveillance Coverage Using Maps Extracted by a Monocular SLAM Algorithm" *International Conference on Robotics and Intelligent System (IROS)*, 2011

- Kneip, L.; Martinelli, A.; Weiss, S.; Scaramuzza, D. & Siegwart, R. "A Closed-Form Solution for Absolute Scale Velocity Determination Combining Inertial Measurements and a Single Feature Correspondence" *IEEE International Conference on Robotics and Automation (ICRA)*, 2011
- Kneip, L.; Weiss, S. & Siegwart, R. "Deterministic Initialization of Metric State Estimation Filters for Loosely-Coupled Monocular Vision-Inertial Systems" *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011
- Voigt, R.; Nikolic, J.; Huerzeler, C.; Weiss, S.; Kneip, L. & Siegwart, R. "Robust Embedded Egomotion Estimation" *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011
- Weiss, S. & Siegwart, R. "Real-Time Metric State Estimation for Modular Vision-Inertial Systems" *IEEE International Conference on Robotics and Automation (ICRA)*, 2011
- Blösch, M.; Weiss, S.; Scaramuzza, D. & Siegwart, R. "Vision Based MAV Navigation in Unknown and Unstructured Environments" *IEEE International Conference on Robotics and Automation (ICRA)*, 2010
- Bourgeois, F.; Kneip, L.; Weiss, S. & Siegwart, R. "Delay and Dropout Tolerant State Estimation for MAVs" *International Symposium on Experimental Robotics (ISER)*, 2010
- Zingg, S.; Scaramuzza, D.; Weiss, S. & Siegwart, R. "MAV Navigation through Indoor Corridors Using Optical Flow" *IEEE International Conference on Robotics and Automation (ICRA)*, 2010
- Steiger, O.; Weiss, S. & Felder, J. "Real Time Understanding of 3D Video on an Embedded System" *17th European Signal Processing Conference (EUSIPCO)*, 2009
- Steiger, O.; Felder, J. & Weiss, S. "Calibration of Time-of-Flight Range Imaging Cameras" *IEEE International Conference on Image Processing (ICIP)*, 2008