DISS. ETH NO. 22113

UNMANNED SOLAR AIRPLANES

Design and Algorithms for Efficient and Robust Autonomous Operation

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

STEFAN LEUTENEGGER

MSc. Mechanical Engineering with Focus Robotics, ETH Zurich
BSc. Mechanical Engineering, ETH Zurich

born on 16.02.1983

citizen of Switzerland and Slovenia

accepted on the recommendation of

Prof. Roland Y. Siegwart (ETH Zurich, Principal Advisor)

Prof. Gerd Hirzinger (DLR, Member of the Jury)

Dr. Kurt Konolige (Industrial Perception, Member of the Jury)

2014

Autonomous Systems Lab
ETH Zurich
Switzerland

© 2014 Stefan Leutenegger. All rights reserved.

To my family and friends.

Acknowledgements

I would like to take the chance of expressing my deepest appreciation and gratitude to Prof. Roland Siegwart, who offered me the unique opportunity of pursuing my doctorate in the Autonomous Systems Lab (ASL): Roland, you have always been able to motivate me and provide valuable inputs while leaving me with the freedom to follow my own ideas. I would furthermore like to thank my postdocs Dr. Paul Furgale for his excellent guidance and ability to challenge me—as well as Dr. Margarita Chli who has been an inspiring adviser unconditionally supporting me also in rough times. You both have been great colleagues and friends!

Moreover, many thanks go to Prof. Gerd Hirzinger for accepting to join the committee. Equally I would like to thank my second external committee member, Dr. Kurt Konolige, who has been supervising me since the time I spent at Willow Garage. Kurt, I will never forget the endless drive home from skiing with all its enriching discussions!

I had the luck of interacting with many scientific and industrial partners in a fruitful manner. I would like to thank Dr. Reiner Kickert, Christian Ückert and Stephan Wrobel for working together on the senseSoar airplane. I am equally grateful for working with Sam Gisiger at Triadis on integrating glider avionic hardware and algorithms. Having had the ASL internal lead of the ICARUS and SHERPA FP7 project has given me the opportunity of interacting with many outstanding and visionary scientists, most notably with the project coordinators Dr. Geert de Cubber and Prof. Lorenzo Marconi, who will continue to be role models for me.

During the past years, it was my pleasure to work with many students at Bachelor and Master level: Mathieu Jabas, Flavio Gohl, Philipp Krüsi, Atilla Yilmaz, Nikolas Gehrig, Fadri Furrer, Péter Fankhauser and Corsin Gwerder, Benjamin Peter, as well as all the focus team "Skye", just to name a few of the many who dedicated their work to this fascinating field of research under my supervision.

My colleagues have made every hour spent working in robotics enjoyable: many thanks go to the "fixed-wing" team consisting of Konrad Rudin, Thomas Mantel, Philipp Oettershagen, Amir Melzer, Dr. Kostas Alexis and our pilot and man for building and repairs, Rainer Lotz. It has been great to work with all of you and much of the progress reported in this thesis has been made thanks to this collaboration! The same holds for the remaining "flying" researchers, Dr. Stephan Weiss, Dr. Laurent Kneip, Dr. Markus Achtelik, Dr. Mike Bosse, Jörn Rehder, Janosch Nikolic, Sammy Omari, Michael Burri and Pascal Gohl. I am furthermore very thankful for the support by the lab engineers Thomas Baumgartner, Stefan Bertschi, as well as the lab administration by Lucy Borsatti and Cornelia della Casa; I am equally grateful for the help by Markus Bühler and Dario Fenner at the workshop, who I challenged repeatedly with parts difficult to manufacture. ASL has been a great work environment, and I do not want to miss thanking every

individual for fruitful collaboration and also the fun times ranging from coffee breaks to lab trips and going out till the morning dawn. I would also like to express special thanks to Dr. Vincent Rabaud at Willow Garage, for mentoring equally as for the fun time spent with other colleagues like Ioan Sucan, Morten Kjægaard, Mihai Dolha, Stefan Holzer, and many more.

My family and friends, to whom I dedicate this work (belonging to this special group of "normal people"), have unconditionally supported me at all times and thus deserve the highest gratitude. René and Bert, I will never forget our legendary 3×30 party! Dear mom, dad and brother: thank you so much for having been there for me at all times!

Zurich, in June 2014.

Abstract

Solar airplanes offer the unique capability of staying airborne for extremely long times: to date, both unmanned and manned systems have proved sustained flight, in the sense of flying through several day and night cycles. During the day, the solar module powers the airplane and re-charges a battery, which must take the airplane through the following night. Small-scale unmanned solar airplanes have thus been suggested for a plethora of non-military application scenarios, ranging from disaster response to Search and Rescue (SaR) as well as general large-scale mapping missions. This thesis addresses many aspects related to long-term autonomous operation of this special class of Unmanned Aerial Systems (UAS) in close proximity to the ground.

We start in the very beginning with asking the question of how large a solar airplane should be and how it would perform, in order to accomplish a target mission. A methodology is presented that performs actual aerodynamics and structural calculations of either a simplified shell or rib wing concept. The performance evaluation part also accounts for flying optimized altitude profiles, in order to allow for potential energy storage. The output of this conceptual design tool has motivated the design of the *senseSoar* solar airplane prototype that is equipped with enhanced sensing and processing components. We describe the details associated with the design of the different components—an engineering effort that spans various disciplines from aerodynamics to solar technology, electronics and avionics as well as structures. Furthermore, we introduce a modular sensing and processing unit that can be attached to a second solar airplane prototype, AtlantikSolar, aimed at record flying. The airplane was developed outside the scope of this work, but its realization was again motivated by the conceptual design tool. Throughout the design process, but also for subsequent simulations and autopilot development, aerodynamics and flight kinematics models play an important role; we present a complete toolchain for such analysis.

Developing *efficient* components is key to any successful solar airplane design. Long-term operation, however, will only be enabled, if the aircraft additionally exhibits sufficient *robustness*. These two central concepts do not only apply to design, but equally to algorithms, which eventually turn the airplane into a system that can operate autonomously. As a basis for any autonomy, the aircraft needs to have an estimate about its internal *states*, as well as about its surroundings, specifically in the form of a *map*. Large parts of the thesis at hand address precisely the associated challenge of fusing various sensor sources under hard real-time and computational constraints.

Specifically, two algorithms are presented and analyzed in detail that share a common element: namely inertial measurements, i.e. accelerometer and rate gyro readings subjected to their kinematics equations. A first fusion strategy complements this inertial module with magnetometer, static and dynamic pressure, as well as GPS measurements that are

combined in an Extended Kalman Filtering (EKF) framework. The approach comes with robustness as it can withstand long-term GPS outage. Respective results show how the crucial states of orientation and airspeed including Angle of Attack (AoA) and sideslip are still tracked sufficiently well in such a case. The latter is mainly enabled by the use of an aerodynamic airplane model. The algorithm was designed to be lightweight, so it can run our microcontroller boards as an input to the autopilot.

A second estimation algorithm complements the inertial module with visual cues: the combination has gained increasing attention lately since it enables accurate state estimation as well as situational awareness. We chose an approach to fuse the landmark reprojection error with inertial terms in nonlinear optimization. The concept of *keyframes* is implemented by resorting to marginalization, i.e. partial linearization and variable elimination, in order to keep the optimization problem bounded—while it may still span a long time interval. The framework can be used with a monocular, stereo or multi-camera setup. As we demonstrate in extensive experiments, our algorithm outperforms a competitive filtering-based approach consistently in terms of accuracy, whilst admittedly demanding more computation. In further experiments, we address calibration of the camera pose(s) relative to the inertial measurement unit; and we validate the *scalability* of the method—which allows to size the optimization window to the available hardware—such that real-time constraints are respected.

In order to address the specific challenge of forming visual keypoint associations, we propose BRISK: Binary Robust Invariant Scalable Keypoints that aims at providing a high-speed yet high quality alternative to proven algorithms like SIFT and SURF. The scheme consists of scale-space corner detection, keypoint orientation estimation and extraction of a binary descriptor string. The latter two steps employ a sampling pattern for local brightness gradient computation and descriptor assembly from brightness comparisons. The evaluation comprises detection repeatability and descriptor similarity as well as timings when compared to SIFT, SURF, BRIEF, and FREAK. The code has been released and has found broad adoption.

The presented platforms with their sensing and processing capabilities are finally used in flight tests to run the suggested algorithms. We show online operation of the inertial navigation filter. Moreover, the stereo-visual-inertial fusion was run on-board a multicopter in the control loop, which has enabled a high level of autonomy. Finally, we demonstrate the application of the mono-visual-inertial algorithm to an AtlantikSolar flight. The algorithm was augmented to accept GPS and magnetometer measurements, outputting airplane state and a consistent map—as a first milestone towards autonomous operations of small solar airplanes close to the terrain.

Keywords: Solar Airplanes, Unmanned Aerial Systems, Airplane Design, Sensor Fusion, State Estimation, Inertial Navigation System, Visual-Inertial Navigation System, Visual-Inertial Odometry, Simultaneous Localization and Mapping, Keyframes, Nonlinear Optimization, Image Keypoints, Image Features, Keypoint Detection, Descriptor Extraction.

Kurzfassung

Solarflugzeuge besitzen die einzigartige Fähigkeit, über extrem lange Zeiträume in der Luft zu bleiben: heutzutage existieren sowohl bemannte als auch unbemannte solcher Systeme, welche bewiesen haben, dass sie kontinuierlich, also mehrere Tag- und Nacht-Zyklen ununterbrochen fliegen können. Dabei versorgt ein Solarmodul während des Tages den Antrieb mit Leistung und lädt gleichzeitig einen Akku, welcher das Flugzeug dann durch die folgende Nacht bringen muss. Davon inspiriert wurden kleine unbemannte Solarflugzeuge für verschiedenste nicht-militärische Anwendungen vorgeschlagen; beispielsweise für den Einsatz in einem Katastrophengebiet, für Such- und Rettungs-Aufgaben oder generell für ausgedehnte Kartierungsanwendungen. Diese Dissertation behandelt mehrere Aspekte des Solarflugs kleiner unbemannter Flugzeugsysteme, welche nahe am Gelände operieren.

Wir beginnen mit der Frage, wie gross denn ein Solarflugzeug auszulegen ist, um eine bestimte Mission erfüllen zu können, und welche Leistungscharakteristiken erwartet werden können. Es wird ein Ansatz vorgestellt, welcher automatisch eine vereinfachte Aerodynamik- und Strukturauslegung durchführt, und zwar wahlweise mittels eines Schalenflügel-Konzepts oder mittels Holm- und Rippenbauweise. Das Modul, welches die Leistungsberechnung anstellt, kann auch optimale Höhenprofile berücksichtigen, welche erlauben, potentielle Energie zu speichern. Die Ausgaben dieses Instrumentes zum Konzeptentwurf hat die Entwicklung des *senseSoar* Solarflugzeugs motiviert, welches mit weitreichender Sensorik und Bordrechnern ausgestattet worden ist. Wir beschreiben die Details des Entwicklungsprozesses welcher diverse Ingenieursdisziplinen überspannt, von Aerodynamik zu Solartechnologie, Elektronik bis hin zu Strukturauslegung. Des weiteren beschreiben wir eine modulare Sensor- und Recheneinheit, welche an einem weiteren Solarflugzeug angebracht werden kann, nämlich an AtlantikSolar. Im Entwicklungsprozess, aber auch für darauf folgende Simulationen sowie für die Auslegung von Autopilotsystemen sind aerodynamische und flugmechanische Modelle von grosser Wichtigkeit; wir stellen eine komplette Werkzeugkette vor für entsprechende Analysen.

Entwicklung *effizienter* Komponenten spielt eine Schlüsselrolle für den Erfolg eines Solarflugzeuges. Langzeiteinsätze können jedoch nur erreicht werden, wenn das Flugzeug zusätzlich genügend *Robustheit* aufweist. Diese beiden Konzepte haben ebenfalls Gültigkeit in bezug auf Algorithmen, welche es schliesslich zu einem selbständig agierenden System machen. Als Basis für jegliche Autonomie muss das Fluggerät sowohl seine internen *Zustandsvariablen*, als auch seine Umgebung kennen, und zwar spezifisch in der Form einer *Karte*. Diese Dissertation behandelt in weiten Teilen genau entsprechende Herausforderungen, nämlich wie verschiedene Sensorsignale unter harten Echtzeitanforderungen und limitierter Rechenleistung verarbeitet werden können.

Spezifisch werden zwei Algorithmen vorgestellt und im Detail analysiert, welche ein Element gemeinsam haben: nämlich Intertialmessungen, also gemessene Beschleunigungen

und Drehraten, die entsprechenden kinematischen Gleichungen gehorchen. Eine erste Strategie zur Fusion komplementiert dieses inertiale Modul mit Magnetometer, statischen und dynamischen Druckmessungen, sowie mit GPS-Messungen, welche im Rahmen eines Erweiterten Kalman Filters (EKF) kombiniert werden. Der Ansatz weist Robustheit auf, weil er lange anhaltenden GPS Ausfällen standhält. Entsprechende Resultate zeigen, dass die essentiellen Zustände der Orientierung wie auch der Geschwindigkeit gegenüber der Luft, inklusive des Anstell- und Schiebewinkels in diesem Fall weiterhin ausreichend genau geschätzt werden können. Letzteres wird hauptsächlich ermöglicht aufgrund der Berücksichtigung der Flugzeugaerodynamik. Der Algorithmus wurde entwickelt, um auch auf Mikrocontroller-Platinen ausgeführt zu werden, als Input für den Autopiloten.

Ein zweiter Schätzer komplementiert das Inertialmodul mit visuellen Informationen: diese Kombination hat in letzter Zeit vermehrt Aufmerksamkeit genossen, da sie sowohl genaue Zustandsschätzung ermöglicht, wie auch die Wahrnehmung der Umgebung. Wir haben einen Ansatz gewählt, welcher Reprojektionsfehler und inertiale Terme im Rahmen einer Nichtlinearen Optimierung zusammenführt. Das Konzept der Schlüsselbilder wurde mittels Marginalisierung realisiert, also der Linearisierung und Variablen-Elimination, um das Optimierungsproblem beschränkt zu halten – wobei es dennoch einen langen Zeitraum berücksichtigen kann. Unsere Software kann sowohl Ein- als auch Zwei- und Mehrkamerasysteme verarbeiten. Wie wir in umfangreichen Experimenten zeigen, übertrifft sie dabei eine kompetitive filterbasierte Vergleichsimplementation durchwegs, wobei zugegebenermassen mehr Rechenleistung benötigt wird. In weiteren Experimenten betrachten wir mitlaufende Kalibrierung der Kamerapose(n) relativ zur Inertialmesseinheit; und wir validieren die *Skalierbarkeit* der Methode – was es erlaubt, das Optimierungsfenster der verfügbaren Rechenleistung und Echtzeitanforderungen anzupassen.

Um die Herausforderung der Assoziation visueller Schlüsselpunkte anzugehen, stellen wir "BRISK" vor (Binary Robust Invariant Scalable Keypoints), womit wir auf eine Alternative hoher Geschwindigkeit abzielen bezüglich etablierter Algorithmen wie SIFT und SURF. Der Ansatz beinhaltet Detektierung von Ecken in Bildkoordinaten und Grössenskala, sowie die Schätzung der Orientierung des Schlüsselpunktes und das Extrahieren eines binären Deskriptors. Letztere zwei Stufen benützen ein Sampling-Muster zur Berechnung des lokalen Helligkeitsgradienten und zum Zusammensetzen des Deskriptors aufgrund von Helligkeitsvergleichen. Die Evaluation umfasst Wiederholbarkeit der Detektion und Ähnlichkeit der Deskriptoren, sowie Zeitmessungen im Vergleich mit SIFT, SURF, BRIEF und FREAK. Der veröffentlichte Code hat breite Annahme gefunden.

Die vorgestellten Platformen mit ihrer Sensorik und ihren Bordrechnern werden schliesslich in Flugtests eingesetzt, um die vorgestellten Algorithmen darauf auszuführen, beispielsweise den Filter zur Inertialnavigation. Ausserdem zeigen wir einen Multicopter, auf dem der stero-visuell-inertiale Schätzer im Regelkreis läuft. Angewendet auf Flugdaten von der Gondel, welche an AtlantikSolar montiert wurde, demonstrieren wir schliesslich den mono-visuell-inertialen Algorithmus, der um die Fähigkeit erweitert wurde, Magnetometer und GPS-Messungen zu verarbeiten: als Resultat werden Zustand und eine konsistente Karte ausgegeben – ein erster Meilenstein in Richtung Einsatz von kleinen Solarflugzeugen nahe am Gelände.

Schlüsselwörter: Solarflugzeuge, Unbemanntes Luftfahrzeug, Flugzeugentwurf, Sensorfusion, Zustandsschätzung, Inertialnavigationssysteme, bildunterstützte Inertialnavigation, Gleichzeitige Lokalisierung und Kartenerstellung, Schlüsselbilder, Nichtlineare Optmierung, Bildschlüsselpunkte, Schlüsselpunterkennung, Deskriptorextraktion.

Contents

1 Introduction **1**
- 1.1 Autonomous Solar Airplane Challenges . . . 2
  - 1.1.1 Application Scenarios . . . 2
  - 1.1.2 Requirements: Design and Algorithms . . . 3
- 1.2 Background . . . 4
  - 1.2.1 Solar Airplane Design . . . 4
  - 1.2.2 Airplane Models . . . 5
  - 1.2.3 Sensor Fusion and Mapping . . . 5
  - 1.2.4 Image Keypoint Detection, Description and Matching . . . 6
- 1.3 Contributions . . . 7
- 1.4 Organization . . . 8

2 Solar Airplane Conceptual Design **9**
- 2.1 Introduction . . . 9
  - 2.1.1 Examples of Solar-Electric Prototypes . . . 9
  - 2.1.2 Conceptual Design Considerations . . . 10
  - 2.1.3 Conceptual Design Tool Overview and Contributions . . . 10
- 2.2 Problem Statement and Overall Operation . . . 11
  - 2.2.1 Endurance and Excess Time . . . 11
  - 2.2.2 Problem Statement, Assumptions and Simplifications . . . 11
  - 2.2.3 Overall Operation . . . 13
- 2.3 Core Module: Scalable Component Models . . . 13
  - 2.3.1 Aerodynamics . . . 14
  - 2.3.2 Power Train . . . 14
  - 2.3.3 Load Cases . . . 14
  - 2.3.4 Structure Dimensioning . . . 15
- 2.4 Performance Estimation . . . 18
  - 2.4.1 Environment Model . . . 19
  - 2.4.2 Flight at Constant Altitude . . . 19
  - 2.4.3 Optimal Variable Altitude Profile . . . 19
- 2.5 Simulation Results and Evaluation . . . 21
  - 2.5.1 Maximum Constant Altitude for Sustained Flight . . . 21
  - 2.5.2 Performance of a Small UAS . . . 23
  - 2.5.3 Low Altitude Record UAS . . . 23
  - 2.5.4 Comparison to Existing Airplanes . . . 26
- 2.6 Discussion . . . 27

3 Solar Airplane Platforms **29**
3.1 Introduction . . . 29
3.1.1 Solar Airplane Design Contributions and Outline . . . 30
3.2 senseSoar . . . 31
3.2.1 System Overview . . . 31
3.2.2 Airframe . . . 34
3.2.3 Power Train . . . 34
3.2.4 Sensing and Processing Electronics . . . 37
3.3 AtlantikSolar . . . 38
3.3.1 System Overview . . . 38
3.3.2 Sensing and Processing Pod . . . 39
3.4 Discussion . . . 40

4 Airplane Dynamic Models **41**
4.1 Introduction . . . 41
4.1.1 Outline and Contributions of the Modeling Methods . . . 43
4.2 Notation and Definitions . . . 44
4.2.1 2D Flow Around Airfoils . . . 44
4.2.2 Forces and Moments Acting on an Airplane . . . 45
4.3 Nonlinear Lifting Line . . . 46
4.3.1 Discretized Representation of the Lifting Surfaces . . . 46
4.3.2 Solving for Induced Velocities, Forces, and Moments . . . 47
4.4 Parametric Nonlinear Model . . . 48
4.4.1 Aerodynamics Model . . . 48
4.4.2 Propulsion Group Model . . . 48
4.4.3 Complete Dynamics Equations . . . 51
4.5 Solar Airplane Characteristics . . . 53
4.5.1 Aerodynamics . . . 54
4.5.2 Aircraft Dynamics . . . 55
4.6 Discussion . . . 56

5 Robust Multi-Sensor State Estimation **57**
5.1 Introduction . . . 58
5.1.1 Related Works . . . 58
5.1.2 Outline and Contributions of the Estimators . . . 62
5.1.3 Filtering Versus Nonlinear Optimization . . . 64
5.2 Prerequisites . . . 65
5.2.1 Notation and Definitions . . . 65
5.2.2 Frames . . . 66
5.2.3 States and Local Parameterization . . . 66
5.3 IMU Kinematics and Bias Dynamics . . . 67
5.3.1 Non-Linear Continuous Time Model . . . 67
5.3.2 Linearized Model of the Error States . . . 68
5.4 Multi-Sensor Fusion without Cameras . . . 69
5.4.1 Estimation Filter Operation . . . 69
5.4.2 Evaluation on Manned Glider Avionics . . . 76
5.4.3 Overview of the Hardware . . . 77
5.5 Keyframe Visual-Inertial Odometry . . . 81

5.5.1 Introduction . . . 82
5.5.2 Visual-Inertial Sensor Setup, Notation and States . . . 84
5.5.3 Batch Visual SLAM with Inertial Terms . . . 85
5.5.4 Frontend Overview . . . 88
5.5.5 Keyframes and Marginalization . . . 90
5.5.6 Results . . . 95
5.6 Discussion . . . 107
5.6.1 Multi-Sensor Fusion without Computer Vision . . . 107
5.6.2 Using Camera Measurements . . . 108

6 BRISK: Efficient Image Keypoints 109
6.1 Introduction . . . 109
6.1.1 Related Work . . . 110
6.1.2 Outline and Contributions of BRISK . . . 111
6.2 Scale-Space Keypoint Detection . . . 112
6.2.1 FAST and HARRIS-Based Corner Score . . . 112
6.2.2 Scale Detection . . . 113
6.2.3 Uniformity Enforcement . . . 114
6.3 Rotation Invariance and Descriptor Extraction . . . 116
6.3.1 Sampling Pattern and Rotation Estimation . . . 116
6.3.2 Building the Descriptor . . . 117
6.3.3 Descriptor Matching . . . 118
6.3.4 Notes on Implementation . . . 118
6.4 Performance Evaluation . . . 119
6.4.1 Detection Repeatability . . . 119
6.4.2 Overall Algorithms Descriptor Similarity . . . 120
6.4.3 Timings . . . 123
6.5 Discussion . . . 124

7 Autonomous Aircraft Flight Testing 127
7.1 Introduction . . . 127
7.2 Impressions, Challenges and Lessons Learned . . . 128
7.2.1 Analysis of the senseSoar Crash . . . 130
7.3 State Estimation Results . . . 132
7.3.1 Multi-Sensor Estimation in the Control Loop . . . 132
7.3.2 Stereo-Visual-Inertial Odometry On-Board Multicopter . . . 138
7.3.3 Extended Visual-Inertial Fusion On Solar Airplane . . . 140
7.3.4 Using GPS Measurements . . . 140
7.3.5 Inclusion of the Magnetometer . . . 141
7.3.6 Reconstruction Results . . . 142
7.4 Discussion . . . 144

8 Conclusion 145
8.1 Results and Contribution . . . 145
8.2 Future Research Avenues . . . 148

List of Figures 150

List of Tables **152**

Bibliography **153**

About the Author **161**

Abbreviations and Acronyms

2D	2-Dimensional
2.5D	2D based, but extended to 3D
3D	3-Dimensional
4D	4-Dimensional
6D	6-Dimensional
AC	Aerodynamic Center
AGAST	Adaptive and Generic Accelerated Segment Test
AoA	Angle of Attack
ASL	Autonomous Systems Laboratory
BA	Bundle Adjustment
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CoG	Center of Gravity
DGPS	Differential Global Positioning System (see GPS)
DoF	Degrees of Freedom
DoG	Difference of Gaussian
EKF	Extended Kalman Filter
ETH	Eidgenössische Technische Hochschule
FAST	Features from Accelerated Segment Test
FPGA	Field-Programmable Gate Array
FoV	Field of View
GigE	Gigabit Ethernet
GPS	Global Positioning System
HALE	High Altitude Long Endurance
IGRF-11	International Geomagnetic Reference Field 2011
IMU	Inertial Measurement Unit
INS	Inertial Navigation System

KF	Keyframe
MAV	Micro Aerial Vehicle
MSCKF	Multi-State Constraint Kalman Filter
NED	North-East-Down
QFF	Q-code for static pressure reduced to mean sea level
RANSAC	RANdom SAmple Consensus
RC	Remote Control
RMS	Root Mean Square
ROS	Robot Operating System
SaR	Search and Rescue
SfM	Structure-from-Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SURF	Speeded Up Robust Feature
SVD	Singular Value Decomposition
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
VINS	Visual-Inertial Navigation System
VIO	Visual-Inertial Odometry
VO	Visual Odometry
WGS 84	World Geodetic System 1984

Notation and Symbols

Throughout this work, we use the following notation and symbols. Note that the list is not exhaustive: some symbols with rather local scope are not listed.

a	A lower-case symbol denotes a scalar (with common capital exceptions).
$\mathbf{a}$	A bold lower-case symbol denotes a vector.
$\boldsymbol{a}$	A bold lower-case italic symbol denotes a homogeneous vector.
$\mathbf{A}$	A bold capital symbol denotes a Matrix.
$\mathbf{1}$	The identity matrix, optionally with dimension as subscript.
$\mathbf{0}$	A zero matrix, optionally with dimensions as subscripts.
$\mathcal{A}$	A set.
$\mathbb{R}$	The Real numbers.
$\mathbb{C}$	The complex numbers.
$\mathbb{R}^3$	The 3D Euclidean space.
S^3	The 3-Sphere group.
$SO(3)$	The 3D rotation group.
$SE(3)$	The Special Euclidean group.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	The Normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.
$E[.]$	The expectation operator.
$\underrightarrow{\mathcal{F}}_A$	A frame of reference in $\mathbb{R}^3$.
${}_A\mathbf{a}$	The vector $\mathbf{a}$ expressed in $\underrightarrow{\mathcal{F}}_A$.
${}_A\mathbf{r}_P$	The position vector from the origin of $\underrightarrow{\mathcal{F}}_A$ to point P expressed in $\underrightarrow{\mathcal{F}}_A$.
${}_A\mathbf{v}_{BC}$	The velocity of $\underrightarrow{\mathcal{F}}_C$ as seen from $\underrightarrow{\mathcal{F}}_B$ represented in $\underrightarrow{\mathcal{F}}_A$.
${}_A\boldsymbol{\omega}_{BC}$	The rotational velocity of $\underrightarrow{\mathcal{F}}_C$ w.r.t. $\underrightarrow{\mathcal{F}}_B$ represented in $\underrightarrow{\mathcal{F}}_A$.
$\mathbf{C}_{AB}$	The rotation matrix transforming a vector ${}_A\mathbf{a} = \mathbf{C}_{AB}\,{}_B\mathbf{a}$.
$\mathbf{q}_{AB}$	A quaternion of rotation.
$\otimes$	The quaternion multiplication, such that $\mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC}$.
$[.]^+$	The left-hand quaternion matrix, such that $\mathbf{q}_{AC} = [\mathbf{q}_{AB}]^+ \mathbf{q}_{BC}$.
$[.]^\oplus$	The right-hand quaternion matrix, such that $\mathbf{q}_{AC} = [\mathbf{q}_{BC}]^\oplus \mathbf{q}_{AB}$.
$[.]^\times$	The cross-product matrix of a vector, such that $\mathbf{a} \times \mathbf{b} = [\mathbf{a}]^\times \mathbf{b}$.
$\boldsymbol{T}_{AB}$	A homogeneous transformation matrix.

ϕ	The WGS 84 latitude.
λ	The WGS 84 longitude.
h	The WGS 84 altitude.
φ	roll angle.
θ	pitch angle.
ψ	yaw angle.
ρ	The density of air.
g	Earth's acceleration due to gravity.
α	The Angle of Attack (AoA) of an airfoil or an airplane.
β	The sideslip angle of an airplane.
Re	The Reynolds number.
c_l, c_d, c_m	The airfoil lift, drag and moment coefficients.
c_L, c_D, c_Y	The airplane lift, drag and side force coefficients.
c_l, c_M, c_N	The airplane roll, pitch and yaw moment coefficients.
b	The airplane wingspan.
A	The airplane wing area.
Λ	The airplane aspect ratio.

Chapter 1
Introduction

Flying robots have become increasingly popular during the past decades, since they address a variety of applications and related research questions. Apart from military use-cases, a plethora of possible civil deployment scenarios have been suggested and successful prototypes have emerged, targeting remote sensing, surveillance, Search and Rescue (SaR), imaging, mapping, and inspection to name a few promising examples.

Here, a focus on solar airplanes will be given, aimed at deployment close to the terrain. Owing to their potential for long-term operation, solar airplanes open up prospectives for missions where no or little operator involvement is needed to perform a potentially large-scale mission such as monitoring, surveillance, victim search or mapping. Figure 1.1 shows the senseSoar solar airplane prototype built as part of this work targeting disaster response, SaR, and mapping at low altitude.



Figure 1.1: The senseSoar solar airplane prototype in flight. Photo: François Pomerleau.

In this respect, the challenges that have motivated the thesis work at hand cover a broad range of engineering and science disciplines—which will be detailed in the following.

1.1 Autonomous Solar Airplane Challenges

Unmanned Aerial Systems (UAS) in general require attention across multiple fields, since they combine challenges from aircraft design and operation with the typical challenges of mobile robotics. We illustrate this intersection of disciplines in Figure 1.2.

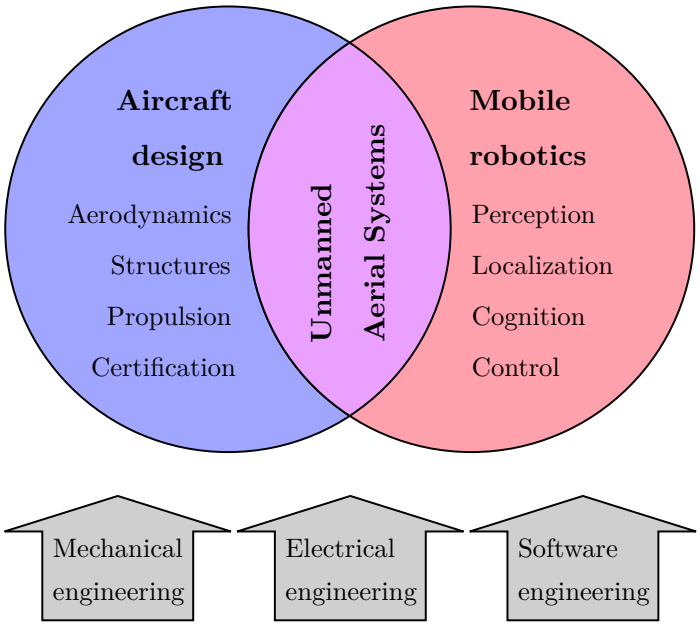


Figure 1.2: UAS design and operation: combined challenges from aircraft design and mobile robotics.

Working with highly autonomous unmanned solar aircraft thus requires insight into aerodynamics and flight mechanics, airplane structures, solar technology, systems integration, avionics and autopilot design, as well as perception and localization.

1.1.1 Application Scenarios

Amongst the many non-military UAS applications suggested and implemented, a summary of the ones most relevant to solar airplanes is given below.

Remote sensing. Measuring parameters both on the ground and in the air in hardly accessible areas has been regarded an interesting use-case of UAS, due to potentially huge savings in terms of resources and emissions when compared to operations with manned aircraft. As examples, wildfire outbreak monitoring and wildlife statistics acquisition have been suggested.

Disaster response. Shortly after the outbreak of a disaster, response teams often lack precise information allowing them to dispatch teams efficiently—since satellite data is typically only available after some days and relies on more or less clear sky. Therefore, the deployment of possibly many UAS in parallel for aerial views, maps and to perform victim search is of high interest. An attempt to provide robotic tools in such settings is

made by the European FP7 project "ICARUS"[1], in which the Autonomous Systems Lab (ASL) participates with its solar airplanes.

Surveillance. Long-term surveillance of borders, pipelines, or facilities in general has been suggested as a further possible application for unmanned solar aircraft: possibly such systems could patrol an extended area without requiring operator workload and thus enable a new level of safety of such objects of interest.

Search and Rescue (SaR). Airborne people search operations are currently mainly performed with manned helicopters. In order to overcome limitations in terms of visibility and low-altitude flight, dedicated UAS have been proposed to complement such missions; the European FP7 project "SHERPA"[2] with ASL as a participant tackles exactly such use-cases.

Image acquisition and mapping. The commercial availability of systems like the small-scale fixed-wing senseFly[3] or multicopters such as those by Ascending Technologies[4] underlines the interest of performing aerial imaging and mapping for various purposes. Solar airplanes could theoretically map virtually unlimited areas.

High-Altitude Long Endurance (HALE). As an attempt to provide an alternative to satellite technology and ground infrastructure targeting e.g. communication, the concept of HALE was suggested: such systems, operating above the clouds, could potentially be equipped with more payload and operated at a lower cost than satellites, and offer more flexibility, since they can be steered. This use-case was approached by a series of NASA solar airplanes, the most recent of which, Helios [Noll et al. 2004], built at a wingspan of 75 meters. Within this thesis, we will, however, focus on low-flying smaller systems, in order to limit complexity concerning design and operational aspects.

1.1.2 Requirements: Design and Algorithms

Solar airplane design and operation comes with the particular demand for long autonomy. This requirement has implications in two domains that are complementary to each-other—namely design of an efficient overall system as well as algorithms and their implementation so that they are lightweight yet accurate enough for on-board deployment.

In essence, all the multi-disciplinary efforts that have to be addressed in order to meet mission-specific requirements are related to the concepts of *efficiency* and *robustness*. An efficient solar airplane design starts at the conceptual stage, where it is sized to optimally comply with the specifications set by the deployment scenario. When realizing the airplane, attention has to be given to efficiency in terms of aerodynamics, but equally by minimizing mass and power consumption of all components. This is illustrated best

[1] ICARUS FP7 project: see www.fp7-icarus.eu, accessed May 2014.

[2] SHERPA FP7 project: see www.sherpa-project.eu, accessed May 2014.

[3] SenseFly: see www.sensefly.com, accessed May 2014. Their "Mapping the Matterhorn" application has drawn much attention.

[4] Ascending Technologies: see www.asctec.de, accessed May 2014.

by stating the overall power consumption:

$$P_{\text{elec,tot}} = \frac{c_{\text{D}}}{c_{\text{L}}^{3/2}} \sqrt{\frac{2(m_{\text{tot}}g)^3}{\rho(h)A}} \frac{1}{\eta_{\text{propulsion}}} + P_{\text{av}} + P_{\text{pld}}. \tag{1.1}$$

The aerodynamics efficiency optimization consists in minimizing the ratio $c_{\text{D}}/c_{\text{L}}^{3/2}$ (with lift coefficient c_{L} and drag coefficient c_{D}). The influence of the total mass m_{tot} is clearly visible; but also the environment, in terms of air density ρ, as a function of altitude h plays a major role. The airplane size, affecting both m_{tot} and the wing area A scales power consumption up and down. Maximizing efficiencies, such as the one of the propulsion group $\eta_{\text{propulsion}}$, is of key importance, not only on the side of output power, as stated in (1.1), but equally on the input side related to the solar power generation. Furthermore, note the terms P_{av} and P_{pld}, denoting avionics and payload power consumption: related efficient algorithms for estimation and control are absolutely necessary, since they limit the power-consumption through the choice of lighter and less power-demanding hardware; this is particularly important for small-scale airplanes, since the related power consumption shares will become more significant the more a design is scaled down. Since time and technology progresses and airplane scales vary, a further desired property of algorithms is their *scalability*—such that they can easily be matched with the available on-board processing power as well as sensors, in order to achieve best *accuracy*.

The concept of robustness is complementary to the need for efficiency, but equally important when it comes to enabling long-term autonomous operation: such solar airplanes are complex systems, each component of which can fail and possibly lead to catastrophic events. It is thus of enormous importance to build robust structures, electronics, software and also algorithms—with necessary redundancy and fail-safe characteristics. The latter includes sensing and state estimation, but extends to motion planning and control, as well as to high-level decision making.

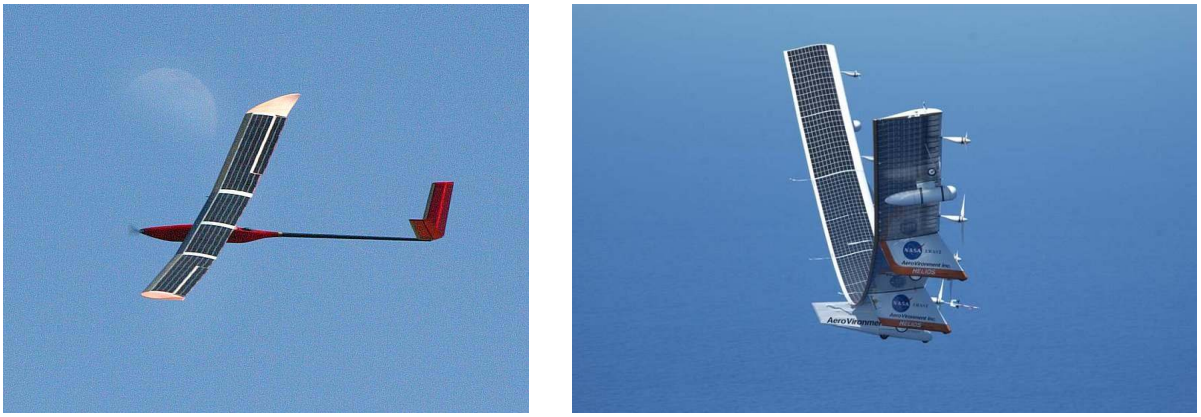
1.2 Background

In order to state the contributions with respect to the state of the art, we provide a short overview of the addressed topics here; a more exhaustive review of related works is given as part of the individual Chapters.

1.2.1 Solar Airplane Design

Solar airplane design starts with finding the main parameters such as wingspan and battery mass, in order to comply with mission-specific requirements; this conceptual design problem has been tackled in different ways. The most relevant work in this respect is the approach by [Noth 2008]. It comes with scalable models for the various components of the airplane, including mass models for the propulsion group and the structure that are derived from statistical data of related existing airplanes. During the past 40 years, a plethora of solar airplanes has been built for various purposes and with continuously improving solar, avionics, and battery technology. It thus comes at little surprise that these designs vary massively in size and achievable performance. A first prototype from 1974, Sunrise I [Boucher 1976], with a wingspan of 10 m, was capable of a few hours of solar flight; today, unmanned *sustained* solar flight (i.e. theoretically eternal—at least in

summer) was proven to be possible by several projects with different wingspans, amongst which are ASL's Sky-Sailor [Noth 2008] at 3.2 m that flew 27 hours, QinetiQ's Zephyr [zep 2006] at 18 m that flew 57 hours, and NASA's Helios at 75 m of wingspan that flew for several day-night cycles. To date, also manned solar flight is at the edge of being feasible in a sustained manner, again with very large designs of extremely light-weight structures—as impressively demonstrated by Solar Impulse [sol 2009]. Figure 1.3 shows the smallest and largest unmanned solar airplane to demonstrate sustained flight.



(a) Sky-Sailor (ASL). (b) Helios (NASA).

Figure 1.3: Two very different examples unmanned solar airplanes that are capable of sustained flight.

1.2.2 Airplane Models

Mathematical models for airplanes play a major role both in the design process as well as for model-based control, simulation and testing in general, and even as part of accurate state estimation. We distinguish between *static* properties provided by aerodynamics alone, and *dynamic* models, i.e. the airplane equations of motion that combine aerodynamics with rigid body dynamics. The topics have been widely discussed in textbooks such as [Etkin 1972] and [McCormick 1979]. 2D, 2.5D and full 3D Computational Fluid Dynamics (CFD) tools are available, partly as freeware. For UAS control, a fully parametric version of the dynamic model needs to be made available—formulated as described e.g. in [Ducard 2009].

1.2.3 Sensor Fusion and Mapping

In order to enable autonomous aircraft operation, real-time state estimation as well as awareness of the environment forms a crucial basis. Fusing inertial measurements with pressure sensors, magnetometers, and GPS has been mostly addressed with filtering-based approaches such as the Extended Kalman Filter (EKF) to form Inertial Navigation Systems (INS). In the *direct* approach, acceleration and rate gyro measurements from the Inertial Measurement Unit (IMU) are used as filter updates, whereas the *indirect* formulation treats IMU measurements as inputs to the underlying kinematics equations. Knowledge about static and dynamic airplane properties can enable estimation of further states such as Angle of Attack, sideslip and wind.

Recently, fusing also visual cues, typically in the form of 2D image keypoint measurements of 3D landmarks has become very popular—an illustration is given in Figure 1.4. This approach exploits the complementary nature of inertial sensing with its strong *temporal* correlations and visual measurements that are of *spatial* nature—while additionally providing information on the 3D surrounding structure. Such Visual-Inertial Navigation Systems (VINS) are mostly filtering based, such as the Multi-State Constraint Kalman Filter (MSCKF) Mourikis and Roumeliotis [2007] that has shown impressive performance below 0.5 % of position error per unit distance travelled, while only exploiting a monocular camera, accelerometers and rate gyros.

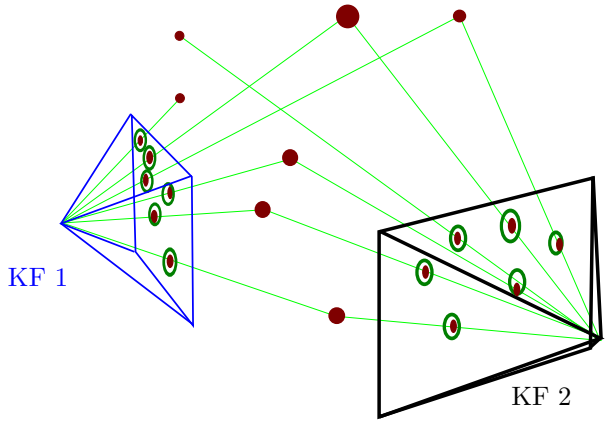


Figure 1.4: Landmark measurements: two cameras associated with robot poses (or keyframes, KF) jointly observe 3D landmarks. Uncertain keypoint measurements (green ellipses) should match the landmark reprojections (small filled ellipses) well, if both the 3D landmark positions and the camera poses are well observed.

Purely vision-based Simultaneous Localization and Mapping (SLAM) historically also employed filtering e.g. by Davison [2003]. More recently, however, methods based on *nonlinear optimization* of the overall reprojection error (see Figure 1.4) of all landmark observations in *keyframes* (Bundle Adjustment) have been shown to offer highest accuracy per computational effort [Strasdat et al. 2010]. Speed is achieved thanks to the sparse nature of the underlying problem; and accuracy owing to re-linearization.

In order to benefit from some of the aforementioned advantages, VINS hybrids such as *fixed-lag smoothers* have been suggested that rely on partial *marginalization*, i.e. the linearization and variable elimination of the sub-part of the nonlinear optimization problem that contains old states. Depending on further approximations, the speed gains due to sparsity may or may not be lost [Dong-Si and Mourikis 2011; Sibley et al. 2010; Huang et al. 2011].

1.2.4 Image Keypoint Detection, Description and Matching

Detecting salient regions in an image and describing them such that they can be matched is a classical computer vision problem. Such a module is needed by estimation algorithms as described above that rely on associated 2D image keypoints. In the most generic setting, such algorithms should be *invariant* to many transformations such as arbitrary scale and rotation changes and should at least be *tolerant* to other distortions as caused by viewpoint change, brightness or contrast variation. This general case is handled well

by established methods such as SIFT [Lowe 2004] or, at higher speeds, by SURF [Bay et al. 2008]. These methods are suited also for place recognition and re-localization; the case of a simple frame-to-frame *tracking* with less change in keypoint position in image coordinates and less visual appearance difference may be handled with different approaches such as the KLT-tracker [Tomasi and Kanade 1991]. Parts of related recent work has focused on increasing speed both at the level of detection, such as with FAST corners [Rosten and Drummond 2006] as well as descriptor matching as by BRIEF [Calonder et al. 2010] that assembles a binary descriptor from brightness comparisons in the keypoint neighborhood. A vision-supported SLAM system ideally has access to fast keypoint detection, tracking as well as re-localization and loop-closure—such that real-time constraints can be met.

1.3 Contributions

The thesis at hand makes contributions across multiple disciplines under the common goal of enabling long-term autonomous solar airplane flight close to the terrain. While this objective is formulated in a quite specific way, it has to be emphasized that most of the individual contributions find much broader applicability in UAS design and operations, mobile robotics and computer vision.

Design of solar airplanes. A conceptual design tool for solar airplanes is presented with the core novelty of calculating actual designs based on (simplified) shell and rib wing structural concepts; in contrast to previous statistics based approaches, it can thus take into account dependencies on payload and battery mass. Results show close agreement with ASL's solar airplanes, that were realized on the basis of this conceptual design methodology. Furthermore, the actual solar platforms built at ASL with various collaborations mark an engineering-level contribution: the senseSoar prototypes as well as the combination of AtlantikSolar with similar sensing and processing capabilities enable the potential for long-term solar flight close to the terrain targeting applications such as disaster response, SaR and mapping. To the best of our knowledge, the availability of such dedicated platforms is novel to the field. Moreover, a whole tool-chain for modeling the static and dynamic characteristics of an electric airplane is presented to enable accelerated detail design, controller synthesis, tuning and simulation in general.

Fail-safe EKF-based state estimation for airplanes A sensor fusion strategy is presented that tightly integrates inertial, pressure, magnetometer, and GPS measurements. The method is proven to be robust to GPS outages over extended amounts of time, which constitutes a substantial advantage over commercially available INS. It furthermore estimates Angle of Attack and sideslip as well as the wind vector by exploiting knowledge on aerodynamic properties of the airplane. The filter was implemented on three different microcontroller boards; it has operated as part of manned glider avionics and it has served as the basis for the autopilot running on all ASL's fixed wing platforms.

Keyframe-based visual-inertial odometry. Inspired from advances of visual SLAM and VINS, we propose a visual-inertial estimation framework that combines IMU error terms and reprojection error in nonlinear optimization: the concept of keyframing is

implemented through the use of marginalization and dropping of observations of non-keyframes. The resulting algorithm can thus track fast and slow motions accurately, while maintaining real-time performance thanks to keeping the sparsity of the underlying normal equations. The framework can integrate one, two or more cameras with an IMU; we describe the necessary implementation and bootstrapping steps. In extensive results, we show the superior performance over a competitive state-of-the-art MSCKF implementation. We furthermore demonstrate online extrinsics calibration as well as scalability of the approach with respect to numbers of keypoints and keyframes.

Efficient image keypoints for real-time applications. We propose BRISK: Binary Robust Invariant Scalable Keypoints as a fast yet high-quality alternative to proven methods of identifying corresponding image keypoint such as SURF and SIFT. Our approach consists of (1) scale-space corner detection with optional uniformity enforcement, (2) orientation estimation by local gradient computation, and (3) the formation of a binary descriptor from brightness comparisons in the keypoint neighborhood (inspired by BRIEF) that are obtained through a carefully designed sampling pattern. In extensive results, we demonstrate the validity of the chosen approach by comparisons to SIFT, SURF, and BRIEF—while timings show significant advantages over SIFT and SURF. The reference implementation was released under BSD license and has been integrated into the OpenCV library[5].

Experiments with unmanned (solar) aircraft. The availability of the unique platforms has enabled a suite of exciting flight experiments. Long-term on-board estimation results include the analysis of a crash with the senseSoar airplane and deployment of stereo-visual-inertial odometry on a multicopter in the control loop. Finally, a solar airplane dataset was processed that contains IMU, magnetometer and GPS measurements as well as imagery; we show that the fusion thereof in a prototyped keyframe-based multi-sensor framework delivers consistent state estimates and a sparse map that will form the basis for autonomous navigation close to the ground and tackling actual applications.

1.4 Organization

The remainder of this dissertation is structured as follows: in Chapter 2, we introduce the methodology employed to conceptually design solar airplanes—such as the senseSoar platform that is described in detail in Chapter 3, along with the design of sensing and processing electronics that may also be attached to ASL's second current solar airplane AtlantikSolar. Some of the modeling techniques that have determined the detail design of our solar airplanes, and that serve as a basis for model-based control, simulation and also play a role in estimation are outlined in Chapter 4. Real-time estimation algorithms with extensive results with and without visual cues are treated in Chapter 5. The particular challenge of how to form image keypoint correspondences is then addressed in Chapter 6 describing and evaluating the BRISK algorithm. Chapter 7 reports on flight testing with the described platforms—including solar airplanes—that run the aforementioned algorithms. Finally, Chapter 8 discusses the major findings and identifies future research avenues.

[5] OpenCV: the Open source Computer Vision library, available under BSD license. See opencv.org, accessed May 2014.

Chapter 2
Solar Airplane Conceptual Design

Solar airplanes exhibit a fascination due to their energy sustainability aspect and the potential for sustained flight lasting several day-night cycles. Resulting monitoring and measurement applications at high altitudes but also close to the Earth surface have attracted the interest of several research groups, institutions and companies lately. The question of how to choose the main design parameters of the airplane for a specific mission, considering the current state-of-the-art technologies involved, however, is not easy to answer. A tool is presented performing such a multi-disciplinary optimization. Solar airplanes using both batteries for energy storage as well as their capability of flying optimal altitude profiles can be sized and evaluated in terms of various performance measures. Simulation results show that sustained flight in the Stratosphere is hard to achieve, if the altitude needs to be kept constant. A simulated Remote Control (RC) model size solar airplane allowed to vary altitude proves to be capable of flying multiple day-night cycles at medium and high latitudes during summer. Finally, suitable designs of a record airplane aimed at crossing the Atlantic Ocean are analyzed. The generated concepts are linked to existing designs at ASL and elsewhere, in order to underline the validity of the chosen approach.

2.1 Introduction

Using sustainable energy sources in aviation, be it manned or unmanned, has increasingly attracted scientists around the world. Solar powered flight started in 1974 with Sunrise I [Boucher 1976]. The prototype designs of solar-electric airplanes which have been built since then are as manifold as their mission characteristics.

2.1.1 Examples of Solar-Electric Prototypes

In 2009, a first prototype of Solar Impulse [sol 2009], took off for the first time: a second prototype rolled out very recently is supposed to take one person around the Earth using nothing but solar energy in 2015. Among the most prominent examples of unmanned

Parts of this chapter appear in:

- S. Leutenegger, M. Jabas and R. Y. Siegwart. Solar Airplane Conceptual Design and Performance Estimation. In *Journal of Intelligent & Robotic Systems 61.1-4*, pages 545–561. 2011.

solar airplanes, there is NASA's High Altitude, Long Endurance (HALE) series with the huge Helios [Noll et al. 2004] (75 m wingspan) as its most recent prototype which proved sustained flight in the stratosphere. This goal was also reached by QuinetiQ's Zephyr [zep 2006] which is much smaller and lighter (18 m of wingspan and weighing 30 kg): in 2010, it flew for more than a week[2]. These HALE platforms exhibit a large potential as low-cost, more flexible alternatives to satellites. But also airplanes that are deployed closer to the Earth's surface offer interesting observation applications such as early wildfire detection, where long endurance provided by solar energy is favorable. Sky-Sailor, the prototype by ETH of medium Remote Control (RC) model size demonstrated sustained flight in the lower atmosphere in 2008 [Noth 2008]. Figure 2.1 shows the aforementioned solar airplanes. A much more exhaustive overview of the history of solar powered flight can be found e.g. in [Noth 2006].



(a) Sunrise II. (b) Helios. (c) Zephyr. (d) Sky-Sailor. (e) Solar Impulse.

Figure 2.1: Some Examples of Solar Airplanes.

2.1.2 Conceptual Design Considerations

When it comes to the conceptual design of solar airplanes, choosing the crucial parameters such as the wingspan that maximize a mission-specific performance measure is a complicated undertaking: the process is highly inter-disciplinary since it involves modeling aerodynamics, environmental characteristics, structural design and assessment of the key technologies such as solar cells and energy storage which undergo rapid progress.

Approaches to such multi-disciplinary optimization are provided by various authors. Already in 1974, Irving and Morgan [1974] assessed the feasibility of a manned solar airplane without energy storage capability and concluded that a plane of 25 m wingspan could sustain several hours. Some years later, Philips [1980] treated both potential and electrical energy storage allowing sustained flight in a NASA technical report outlining also the technological advances necessary.

2.1.3 Conceptual Design Tool Overview and Contributions

We present a tool for the conceptual design of solar airplanes implemented in Matlab, with the following core characteristics.

- In order to find the critical design parameters of wingspan, aspect ratio and battery mass to be carried, it allows simulating missions with various constraints and outputs performance measures which can be optimized. It employs models compliant with existing tools such as found in [Noth 2008], but fundamentally differs in the way of modeling the structural mass: rather than basing estimates on statistics, the

[2] See www.qinetiq.com/media/news/releases/Pages/zephyr-2010.aspx, accessed May 2014.

main design calculations of real feasible lightweight (but simplified) structures are carried out. This implies that mass distributions of scientific payload, propulsion group and energy storage devices are take into account, since they largely influence the necessary structural mass.

- Energy storage in the form of batteries as well as in the form of altitude is addressed. The derivation of optimal altitude profiles is carried out with respect to the aforementioned parameters.

- We apply the tool for sizing UAS to three exemplary missions, namely a small airplane for low altitude monitoring, a High-Altitude Long Endurance UAS (HALE), as well as a record-airplane for low altitude. The sizing and performance estimates are compared to the characteristics of solar airplanes built at ASL, i.e. Sky-sailor, senseSoar and AtlantikSolar.

The remainder of this chapter is organized as follows: in Section 2.2, an overview of the overall operation of the tool is provided and the meaning of the performance measures is introduced. Section 2.3 explains the components of the design tool, in particular the structure dimensioning. Section 2.4 addresses the performance estimation module. Finally, example results obtained with the tool are shown in Section 2.5, discussed, and compared to existing solar airplanes.

2.2 Problem Statement and Overall Operation

Here, we first introduce performance measures that are needed for the subsequent problem statement and overview of the tool operation.

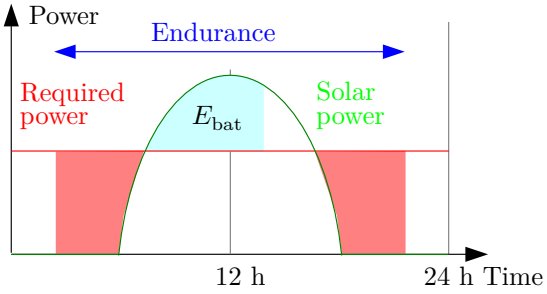
2.2.1 Endurance and Excess Time

When evaluating the performance of a solar airplane, two different results may be obtained: the aircraft can either be capable of flying theoretically eternally or not (disregarding change of day duration as the season changes). Figure 2.2 illustrates these cases and also the direct influence of solar power available, total electric required power and battery energy capacity ($E_{\text{bat,max}}$). For simplification, Figure 2.2 only treats flight at constant altitude.

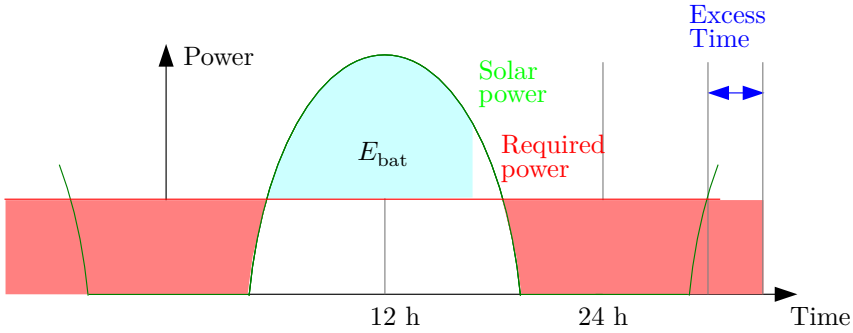
If sustained flight is impossible, the maximum *endurance* (T_{endur}) will be a reasonable performance measure. However, if it is possible to fly several days, a different measure has to be defined: the *excess time* (T_{excess}) is the time the airplane could still fly in complete darkness after one successful day-night cycle. This measure can therefore be regarded as a safety margin. Notice that $T_{\text{endur}} > 24$ h does not necessarily imply that sustained flight is possible.

2.2.2 Problem Statement, Assumptions and Simplifications

The problem can now be stated as follows: given certain environmental and technological parameters, we want to determine the wingspan b, the aspect ratio Λ and the battery mass m_{bat} that optimize a certain performance measure such as the excess time T_{excess} or the endurance T_{endur} of the solar airplane to be designed.



(a) Sustained flight impossible: the maximum endurance is calculated.



(b) Sustained flight possible: the excess flight time in the morning is calculated.

Figure 2.2: The influence of solar power, required power and the battery capacity for the two cases where in 2.2a sustained flight is not possible, but in 2.2b it is. As related performance measures, we introduce *endurance* and *excess time*.

Thereby, some assumptions and simplifications are made. Note that the technological parameters can be adapted as progress is made.

- In terms of electrical energy storage, only rechargeable batteries are considered featuring a gravimetric energy density which does not depend on the capacity. Right now, Lithium-Ion Batteries are storing approximately 220 Wh/kg, which is the highest value of commercially available cells. The development of higher energy density battery technology is ongoing, e.g. with Lithium-Sulfur cells by Sion Power[3]

- Two simplified structure variants are considered—a rib wing and a shell wing concept that are explained in detail in Section 2.3.4. These concepts are applicable to lightly loaded wings in the span ranges not far over 10 meters. Beyond these values, the correctness of the estimates is supposed to decrease since more efficient structural concepts may be chosen.

- Solar cells of highest power-per-weight ratio are chosen and not varied in the optimization, such as by Azur Space[4]. An efficiency value of 19% is taken at a mass density of 420 g/m^2 including encapsulation. Furthermore, the whole wing surface area is assumed to be horizontal and covered completely with solar cells.

[3] See www.sionpower.com, accessed May 2014.
[4] See www.azurspace.com, accessed May 2014.

2.2.3 Overall Operation

Due to the complexity and links between the different components, the tool searches iteratively for optima: the interesting parameters for optimization are clearly the wingspan, the aspect ratio and the battery mass. Within one loop, it evaluates the performance of a given configuration. Figure 2.3 depicts the different components and the respective inputs and outputs. The Core Module inside itself needs to iteratively solve for aerodynamics, power train and structure dimension—which is described in detail in section 2.3.4.

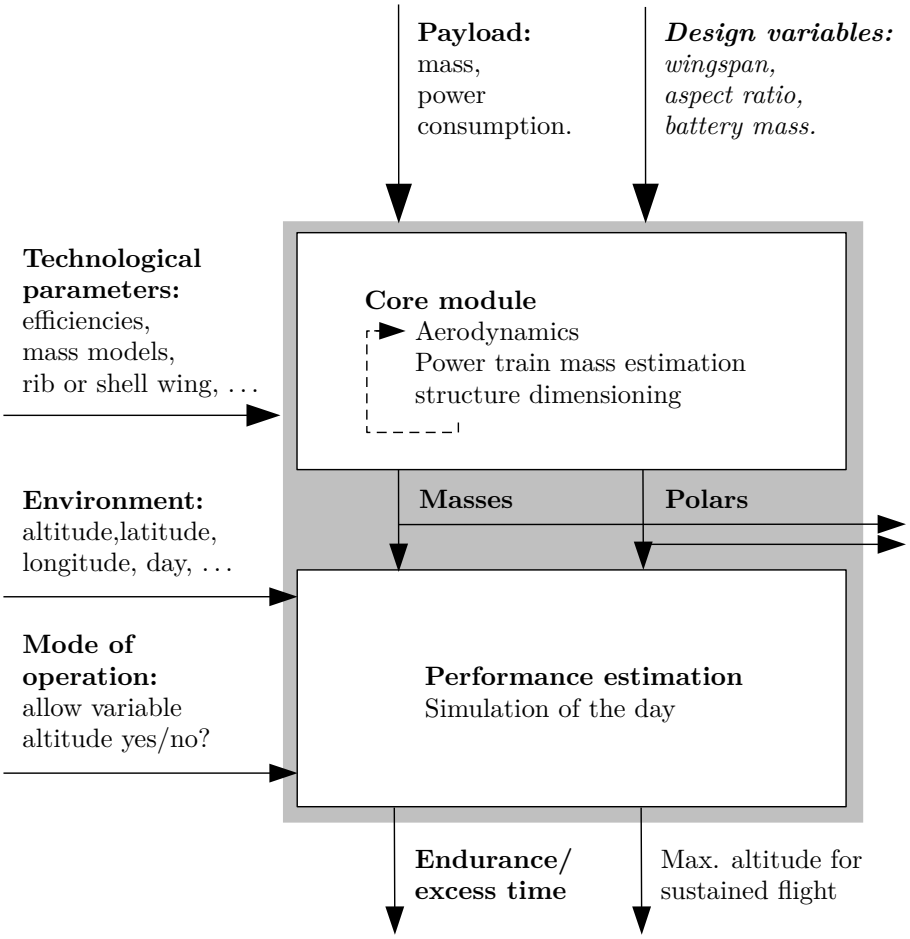


Figure 2.3: Overall operation of the design tool (one iteration): the italic design variables are to be optimized.

This module can now be integrated into a standard optimization framework such as MATLAB's function `fminsearch`, in order to minimize a certain cost function with respect to some of the input variables.

2.3 Core Module: Scalable Component Models

The Core Module operates as a fix-point iteration on aerodynamics and component masses: It starts with an initial guess of structure mass using the statistical prediction as described below in Section 2.3.4. Next, the aerodynamics associated with different load cases as well as the propulsion group mass are calculated allowing to derive the maximum

Shear/Moment/Torque (SMT) loads applied to the various components. This allows calculating the minimum necessary thicknesses of structural elements: both a simplified rib wing and shell wing concept are considered. Finally, the new mass distributions are calculated and the next iteration starts. In the following, the aforementioned sub-modules are described in detail.

2.3.1 Aerodynamics

For simplification and constraints of computational power, the airfoil is not varied: the MH139F designed for low power airplanes by Dr. Martin Hepperle was used, the polars of which were obtained using xfoil[5] for a wide range of Reynolds numbers—but any other airfoil can in principle be employed by the tool. For fast processing, the lift distribution on the main wing is approximated using Schrenk's Method [Schrenk 1940]. For the performance evaluation, the induced drag is still estimated as:

$$c_{D,\text{ind}} = \frac{k c_L^2}{\pi \Lambda}, \tag{2.1}$$

where Λ is the aspect ratio and $k = 1.08$ is assumed in order to correct for non-elliptical lift distribution.

Fuselage and stabilizer drag is approximated using flat plate friction with

$$c_f = 0.074 Re^{-0.2}. \tag{2.2}$$

At the stabilizers, a conservative assumption of uniform force distribution is made for the loading calculation.

As the point of operation, we choose the minimum power point $c_L^3/c_D^2|_{\text{min}}$. Note that the corresponding airspeed typically lies rather close to the stall—making it difficult to track it in practice.

2.3.2 Power Train

For the dimensioning and weight estimation of the power train, a guess for the necessary power is made using a reasonable choice of the required climbing angle α_{cl} (between 20° for hand launched models and 10° for very large airplanes) at climbing speed v_{cl}:

$$P_{\text{max}} = \frac{1}{\eta_{\text{propulsion}}} \left(v_{\text{cl}} m_{\text{tot}} g \sin \alpha_{\text{cl}} + P_{\text{level}} \right), \tag{2.3}$$

where $\eta_{\text{propulsion}}$ denotes the propulsion efficiency. As described in [Noth 2008], an estimate for the propulsion group mass $m_{\text{propulsion}}$ based on statistics can now be obtained using

$$m_{\text{propulsion}} \approx 0.0011 \text{ kg/W } P_{\text{max}}. \tag{2.4}$$

2.3.3 Load Cases

On the basis of the European Aviation Safety Agency (EASA) regulations for gliders and motor gliders CS22 [cs2 2003] as well as on the Unmanned aerial vehicle Systems Airworthiness Requirements (USAR) [usa 2009], loads are defined which the structures have to bear. The following operation points define the respective load cases:

[5] Available free of charge at web.mit.edu/drela/Public/web/xfoil as of March 2014.

- Maximum positive and maximum negative load factors $n_{\max}$ and $n_{\min}$ at manoeuvre speed v_{m}:

$$n_{\max} = 2.1 + \frac{10900 \text{ kg}}{m_{\mathrm{tot}} + 4536 \text{ kg}}. \tag{2.5}$$

- $n_{\max}$ and $n_{\min}$ at dive speed $v_{\mathrm{d}} = 1.5 v_{\mathrm{m}}$.
- ± 7.5 m/s gusts at dive speed
- Full aileron deflection (20°) at manoeuvre speed
- Full elevator deflection (30°) at manoeuvre speed
- Full rudder deflection (30°) at manoeuvre speed

The resulting Shear/Moment/Torque loads (SMT) are determined considering the respective mass distribution and load factors as well as roll/pitch/yaw angular accelerations.

2.3.4 Structure Dimensioning

In the past, several different scaling laws for airplane structures were suggested. Most of them are based on statistics and depend only on a few inputs such as wingspan and aspect ratio Λ. In the design methodology [Brandt and Gilliam 1995], for instance, a proportional relationship of surface (wing, fuselage wetted surface, etc.) and corresponding weight is taken. The approach of assuming a constant wing loading is also chosen in [Rehmet et al. 1997] and other design considerations. Looking at the whole scale range of airplanes, a cubic relationship of weight and scale becomes apparent which is illustrated in Figure 2.4 showing Tennekes' Great Flight Diagram [Tennekes 1992] which has been augmented with RC sailplanes and unmanned solar airplanes based on [Noth 2008]. Notice the tremendous statistical variance making it conceptually impossible to derive a precise model based on only few size parameters.

Based on statistics of the lightest 5% of RC model gliders and manned gliders, Noth [2008] also found an almost cubic relationship:

$$m_{\mathrm{structure}}(b, \Lambda) = \frac{0.44 \text{ N}}{g} \left(\frac{b}{\mathrm{m}} \right)^{3.1} \Lambda^{-0.25}. \tag{2.6}$$

This will be used as a starting guess of structural weight by the structure component of the tool.

The methodology described in short here is comparable to [Hall and Hall 1984] and [Berry 2000], where a real, simplified structure is dimensioned. Mass distributions and load cases are used and heavily influence the resulting structural weight. The downside of this approach is its computational complexity, but offers a significant increase of precision.

Rib Wing and Shell Wing Dimensioning

As stated above, the tool calculates two simplified wing structures: cross-sections of a simplified rib wing concept and a shell wing structure are illustrated in Figure 2.5, where c is the wing chord length.

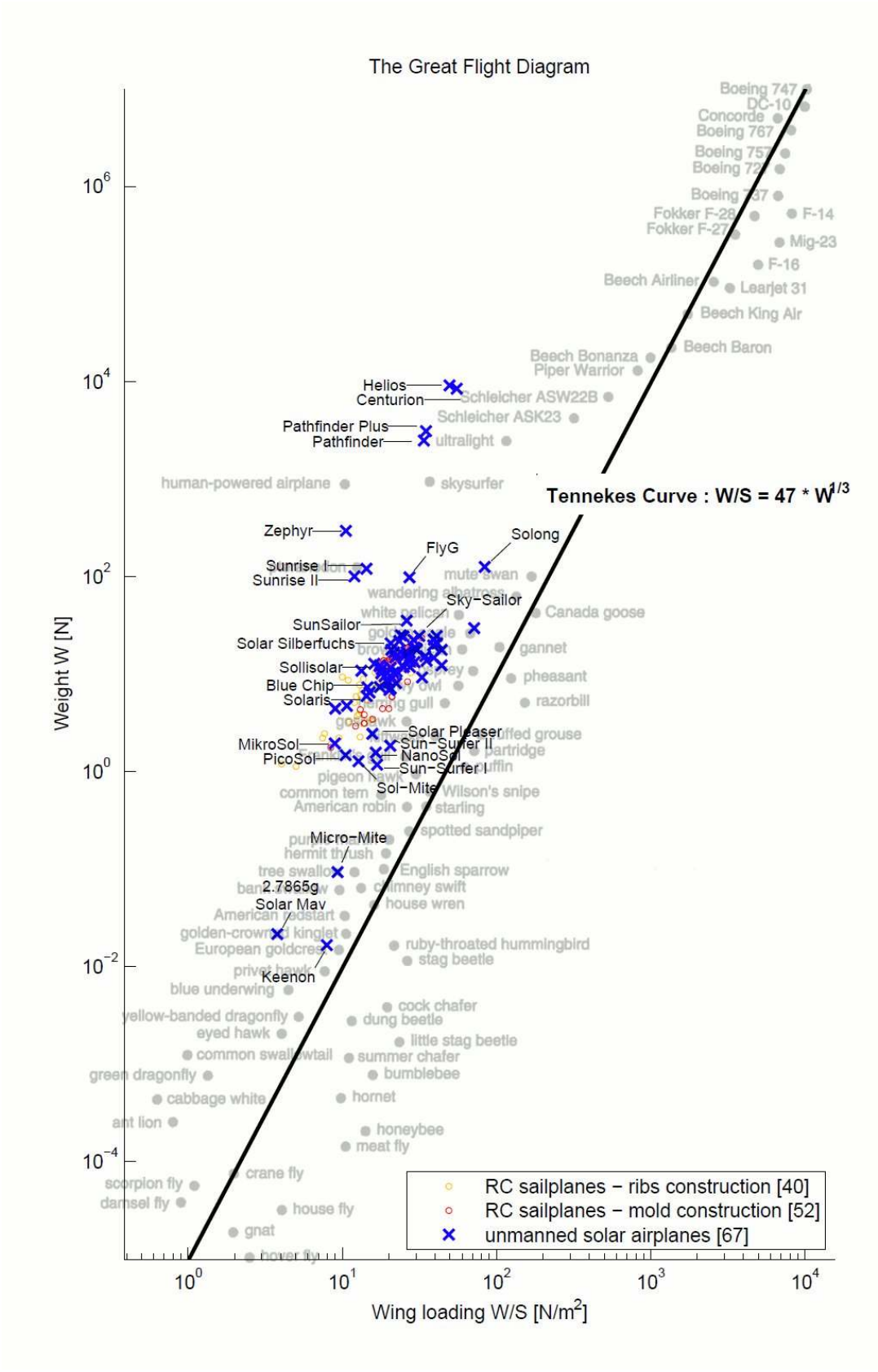
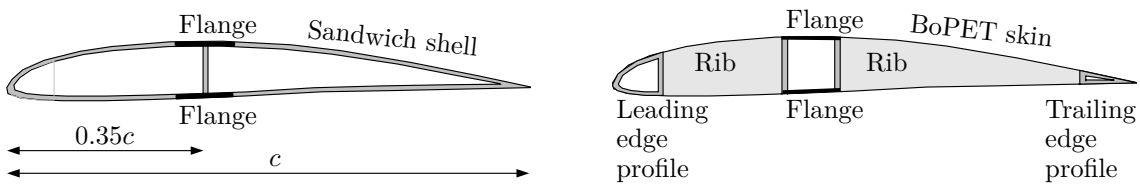


Figure 2.4: Tennekes' Great Flight Diagram augmented with some RC sailplanes and unmanned solar airplanes using data collected in [Noth 2008].



(a) Shell wing concept.

(b) Rib wing concept.

Figure 2.5: Two structure variants available for structural computation in the conceptual design tool.

The respective thicknesses are dimensioned such that the various specified safety factors and maximum deflections are matched exactly and for a minimal mass. This calculation can be done analytically.

Many practical problems arise before being able to do the calculations associated with the dimensioning, to name the most obvious ones:

- A materials database had to be collected. For the case at hand, measured yield strengths, Young's and flexular moduli for composite materials are used.
- Ratios such as between wing, tail units, fuselage length and also the spacing between the ribs have to be defined. Recommendations are found in airplane design literature.

The automatic structural sizing follows textbook guidelines, mostly found in Hertel's "Leichtbau" [Hertel 1960]. Table 2.1 lists the main criteria obeyed, each of them yielding associated minimum component thicknesses, of which the largest one has to be selected while still respecting minimum laminate thicknesses. It is noteworthy that both a carbon

- Bending of the spar flanges: max. 10° deflection, compression buckling, yield.

Specific to Shell Wing:

- Shear in the spar web: yield, shear buckling.
- Torsion in the shell: shear buckling, yield, max. 3° twist.

Specific to Rib Wing:

- Combined shear force and torque in the spar shell: shear buckling, yield, max. 3° twist.
- Ribs compression and shear buckling due to aerodynamic forces and skin tension.
- Shear buckling, max. displacement and yield of leading and trailing edge profiles.

- Fuselage bending: buckling, yield, max. deflection 2°.
- Stabilizers: identical to main wing.

Table 2.1: Main criteria determining the airplane structure.

fiber reinforced (CFR) solution and a glass fiber reinforced version (GFR) is calculated for some components, such that the lighter one can be chosen: at small scale, GFR can in fact be superior, due to thinner available sheets.

Providing all equations for the calculation of these different cases would go beyond the scope of this chapter. Therefore, just the prototype example of shear buckling in the shell is given—other buckling cases are treated in similar manner. Hereby, the thickness of the inner and of the outer laminate $t_{\mathrm{s,tot}}$ and the sandwich core material thickness t_{sw} are determined.

First, the maximum torque $T_{\max}$ is extracted from the different load cases. Next, the shearflow is calculated according to the Bredt-Batho relation:

$$\tau t_{\mathrm{s,tot}} = \frac{T_{\max}}{A_{\mathrm{c}}}, \tag{2.7}$$

where A_{c} is the profile cross-section area. The critical stress for non-curved surfaces is obtained with

$$\tau_{\mathrm{crit}} = kE_{\mathrm{corr}}\kappa \left(\frac{t_{\mathrm{s,tot}} + t_{\mathrm{sw}}}{0.65c}\right)^2, \tag{2.8}$$

where $\kappa = 3$ for symmetrical sandwiches, $E_{\mathrm{corr}} = E_{1,2,\mathrm{lam}} t_{\mathrm{sw}}/(t_{\mathrm{s,tot}} + t_{\mathrm{sw}})$ is the overall sandwich elastic modulus based on the laminate modulus $E_{1,2,\mathrm{lam}}$, $0.65c$ denotes the longest (almost non-curved) width occurring in the shell and $k = 4.8$ is the buckling factor for the boundary conditions at hand. With a safety factor $S_{\mathrm{F}} = 1.5$, we require $\tau S_{\mathrm{F}} = \tau_{\mathrm{crit}}$. Since the structure should be of minimum weight, $\rho_{\mathrm{lam}} t_{\mathrm{s,tot}} + \rho_{\mathrm{core}} t_{\mathrm{sw}}$ is minimized analytically, with ρ_{lam} and ρ_{core} being the fiber reinforced polymer density, the core material density, respectively.

The resulting equations for the thicknesses are:

$$t_{\mathrm{s,tot}} = \sqrt[3]{\frac{\rho_{\mathrm{core}} S_{\mathrm{F}} T_{\max} (0.65c)^2}{\rho_{\mathrm{lam}} 2A_{\mathrm{c}} k\kappa E_{1,2,\mathrm{lam}}}} \tag{2.9}$$

$$t_{\mathrm{sw}} = \frac{S_{\mathrm{F}} T_{\max} (0.65c)^2}{2A_{\mathrm{c}} k\kappa E_{1,2,\mathrm{lam}} t_{\mathrm{s,tot}}^2}. \tag{2.10}$$

As stated before, the minimum feasible laminate thicknesses must be respected at least for the outer laminate; an inner laminate can be replaced by appropriately spaced rovings allowing to achieve an arbitrarily low thickness.

2.4 Performance Estimation

The performance of a configuration is measured as endurance (T_{endur}) or excess (T_{excess}) time as explained above. In some cases, the application may require the solar airplane to fly at constant altitude, where the performance evaluation is pretty straight-forward and forms a basis. The respective performance values are obtained by simulation of the battery state-of-charge evolution during around a day. If, however, altitude variation is permitted as a means of improving performance due to potential energy storage capability, the respective dynamic optimization problem needs to be solved.

2.4.1 Environment Model

The solar irradiation is modeled as a function of geographic location (latitude ϕ), altitude h, time t and surface normal direction according to Duffie and Beckman [1980].

The temperature and air density are estimated using the International Standard Atmosphere. The dynamic viscosity is determined with Southerland's Formula.

2.4.2 Flight at Constant Altitude

The basics for any considerations is the power needed for level flight at altitude h:

$$P_{\text{level}} = \frac{c_{\text{D}}}{c_{\text{L}}^{3/2}} \sqrt{\frac{2(m_{\text{tot}} g)^3}{\rho(h) A}}. \tag{2.11}$$

The electrical power required is derived using several efficiencies (of propeller η_{prop}, motor with gearbox η_{mot} and motor controller η_{ctrl}). Finally, the power consumed by avionics P_{av} and by the payload P_{pld} is added:

$$P_{\text{elec,tot}} = \frac{P_{\text{level}}}{\eta_{\text{prop}} \eta_{\text{mot}} \eta_{\text{ctrl}}} + P_{\text{av}} + P_{\text{pld}}. \tag{2.12}$$

On the side of the inflow, the solar power available is derived as a function of the irradiance $I(h, \phi, t)$ as follows:

$$P_{\text{solar}} = I \eta_{\text{sc}} \eta_{\text{cbr}} \eta_{\text{mppt}}, \tag{2.13}$$

where η_{sc} and η_{mppt} are the solar cell and Maximum Power Point Tracker (MPPT) efficiencies, respectively. η_{cbr} accounts for solar module level losses mainly caused by camber of the cell arrangement. These losses are typically in the order of 10% [Noth 2008].

Knowing incoming and spent power, it is straightforward to derive the battery energy state differential equation. It has to be noted that charge and discharge efficiencies are taken into account.

As an alternative performance measure to excess time (T_{excess} of sustained flight at a certain altitude, the maximum altitude for sustained flight can be determined. This is done by a evaluating the performance as described above as a search in the altitude range between 0 and 30 km Above Mean See Level (AMSL).

2.4.3 Optimal Variable Altitude Profile

Already in [Philips 1980], altitude profiles are discussed. However, the profiles suggested here look slightly different as will be seen below.

In order to formulate the problem as an optimal control problem with input $u \geq 0$ being the electric power sent to the motor, the dynamics equations can be represented, slightly simplified, in the following form:

$$\frac{dE_{\text{bat}}}{dt} = P_{\text{solar}}(h, t) - u - P_{\text{av}} - P_{\text{pld}}, \tag{2.14}$$

$$\frac{dh}{dt} = \frac{\eta_{\text{propulsion}} u - P_{\text{level}}(h)}{m_{\text{tot}} g}, \tag{2.15}$$

with the initial state at t_0 corresponding to the power equilibrium $P_{\text{level}} = P_{\text{solar}}$:

$$h(t_0) = h_0, \tag{2.16}$$
$$E_{\text{bat}}(t_0) = 0. \tag{2.17}$$

The following state constraints are imposed:

$$h(t) \geq h_{\min}, \tag{2.18}$$
$$E_{\text{bat}}(t) \in [0, E_{\text{bat,max}}]. \tag{2.19}$$

The solar power is forced to still be zero after the night, in order to be able to formulate the problem easily with the partly constrained final state at a fixed t_{end} (not hurting the state constraints):

$$h(t_{\text{end}}) = h_0. \tag{2.20}$$

And the cost functional to be minimized can be formulated as:

$$J(u(t), h(t)) = -\int_{t_0}^{t_{\text{end}}} \dot{E}(t)\, dt. \tag{2.21}$$

This can be transformed into

$$\begin{aligned} J(u(t), h(t)) &= \int_{t_0}^{t_{\text{end}}} \Big(u - P_{\text{solar}}(h,t) + P_{\text{av}} + P_{\text{pld}}\Big)\, dt \\ &= \int_{t_0}^{t_{\text{end}}} \left(\frac{\dot{h}(t) m_{\text{tot}} g}{\eta_{\text{propulsion}}} + P_{\text{level}}(h) - P_{\text{solar}}(h(t),t) + P_{\text{av}} + P_{\text{pld}}\right) dt. \end{aligned} \tag{2.22}$$

Since the first part is obviously 0, because $h_0 = h_{\text{end}}$ and due to the avionics and payload power consumption P_{av} and P_{pld} being constants, the following simplified cost integral is obtained:

$$\tilde{J}(h(t)) = \int_{t_0}^{t_{\text{end}}} P_{\text{level}}(h(t)) - P_{\text{solar}}(h(t),t)\, dt. \tag{2.23}$$

Notice that $\tilde{J}$ is only implicitly depending on the control input u. Furthermore, it can be easily verified that

$$\frac{\partial (P_{\text{level}}(h) - P_{\text{solar}}(h,t))}{\partial h} > 0. \tag{2.24}$$

is valid for all reasonable airplane parameter choices in clear sky conditions. Therefore, the problem can be regarded as always flying as low as possible while still obeying the state constraints. The corresponding optimal trajectory can be formulated easily even as a feedback law:

- Start at $h_0 = h_{\min}$ at the power equilibrium $P_{\text{solar}} = P_{\text{level}} + P_{\text{av}} + P_{\text{pld}}$.
- Fly at the minimum altitude $h_{\min}$. The battery is being charged.
- If $E_{\text{bat}} = E_{\text{bat,max}}$ is reached, choose $u = P_{\text{solar}}(h,t) - P_{\text{av}} - P_{\text{pld}}$, i.e. start climbing using all excess power but not more; the battery stays full.
- As soon as $P_{\text{solar}}(h,t) - P_{\text{av}} - P_{\text{pld}} < 0$, descend with $u = 0$ (gliding) down to $h_{\min}$.
- Fly level at $h_{\min}$ until the battery is empty.

A sample profile of a small solar airplane as described in 2.5.2 can be found in Figure 2.6.

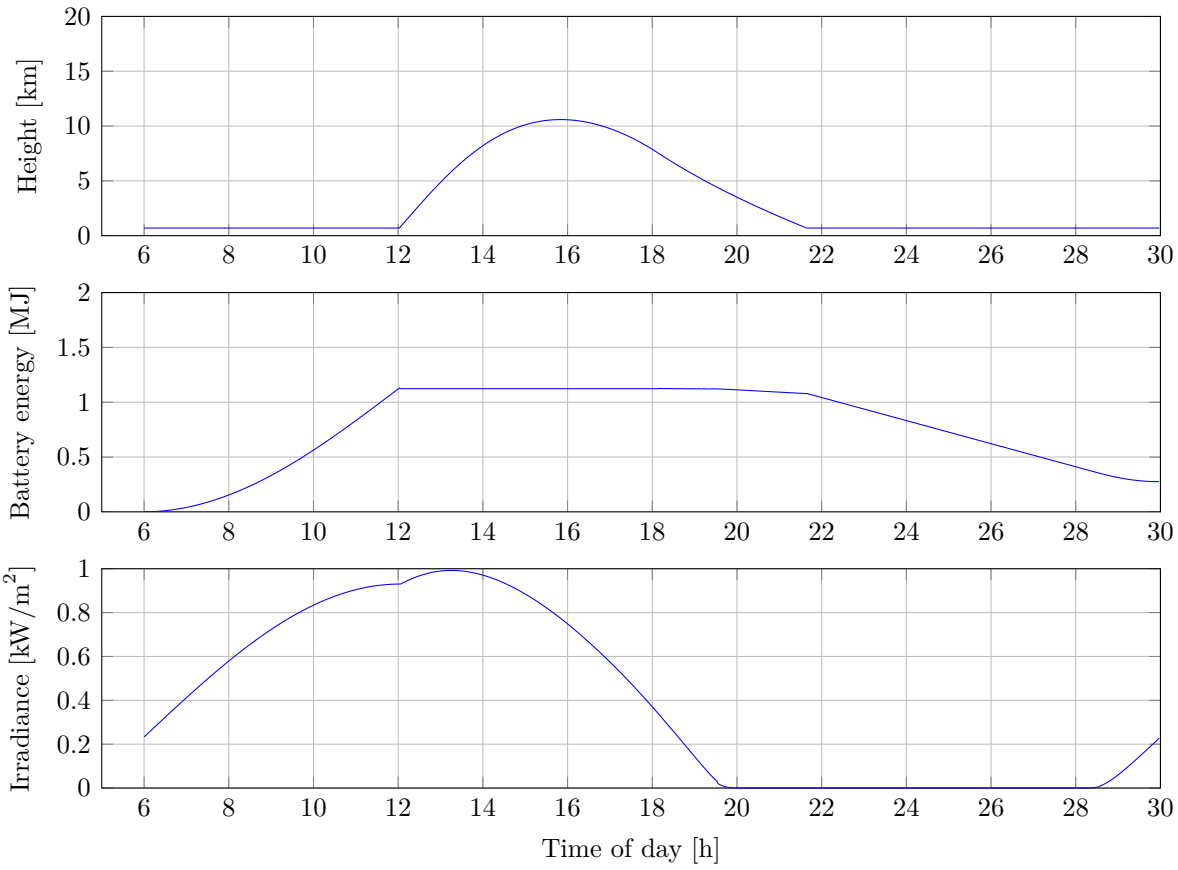


Figure 2.6: Altitude, battery energy and irradiation profile of a small solar airplane carrying minimal payload. It can be observed that sustained flight is possible with this airplane in these conditions.

2.5 Simulation Results and Evaluation

Here, we analyze example applications and concepts found with the presented tool. Furthermore, we relate the outputs of our algorithm to existing airplane designs, two of which were in fact employing the tool at the conceptual design stage.

2.5.1 Maximum Constant Altitude for Sustained Flight

A first interesting question that can be answered with the presented tool is the following: how high could a solar airplane fly continuously, during several day-night cycles as a function of latitude given a certain season? And what are the design parameters needed to achieve this? What are the resulting speeds and are these airplanes going to be able to fly above the tropopause where the sun is not blocked by clouds?

Obviously, the answer to these questions is highly dependent on the payload requirements. A mass of 0.6 kg and a continuous power consumption of 4 W is assumed. This is about the minimum necessary to allow the use of a thermal camera and a minimal data link to a ground station. The rib wing concept is employed, since it will generally result in lower structural masses than with a shell wing.

Figure 2.7 summarizes the answers to the questions asked above for the optimal case of summer solstice.

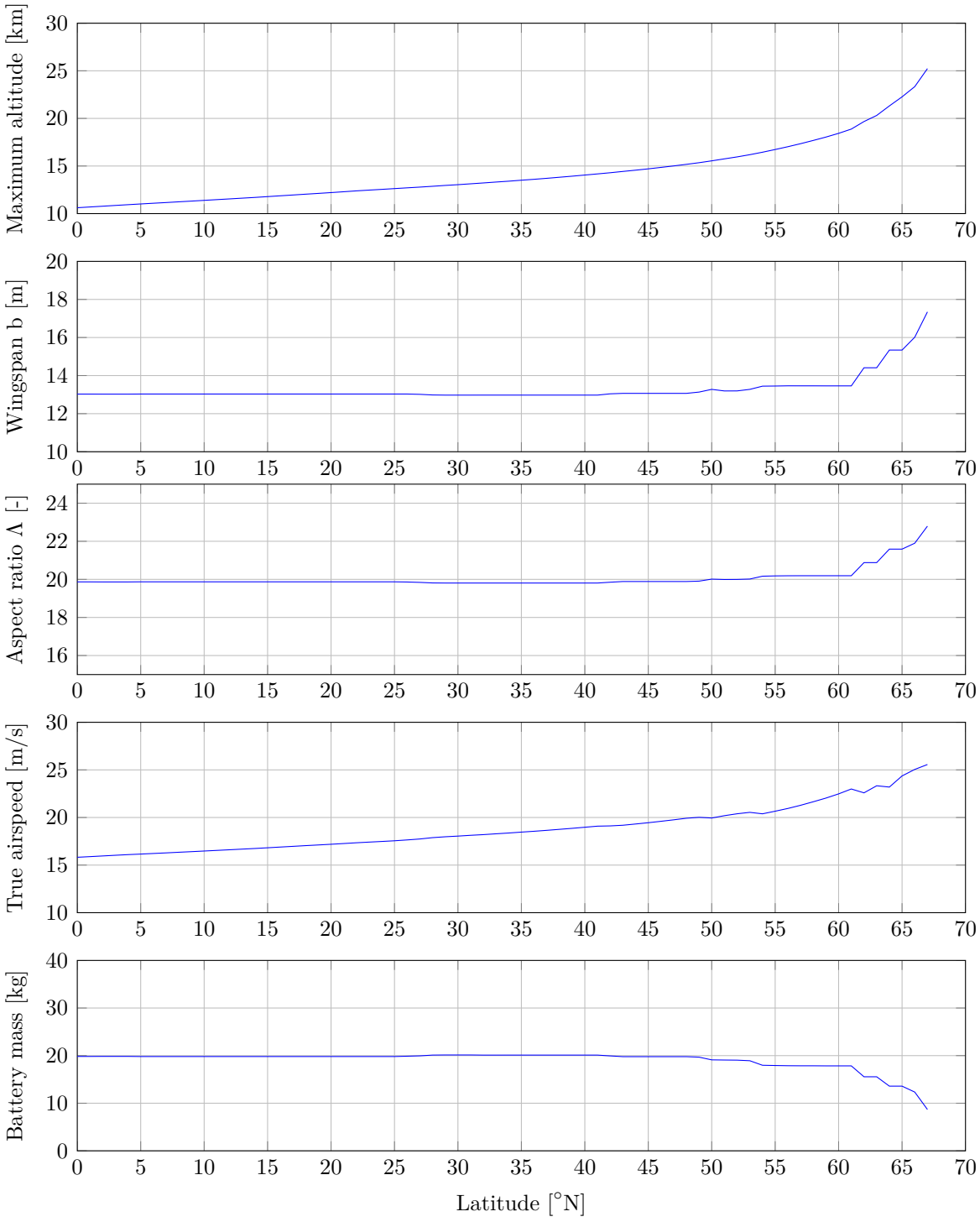


Figure 2.7: Maximum constant altitude reachable in sustained flight at summer solstice; corresponding wingspan, aspect ratio, battery mass as well as true airspeed.

The following points are worth emphasizing:

- The maximum altitude increases smoothly with latitude, but values higher than the tropopause are only achieved north of approximately 40° N.

- A wingspan b around 13 m and an aspect ratio Λ of approximately 20 seems to be most suitable throughout the most relevant band of latitudes.
- The optimal battery mass interestingly also does not vary dramatically as a function of latitude: 20 kg appears to be suitable for most latitudes.
- The true airspeeds of 15 to 20 m/s are below typical wind speeds in the lower Stratosphere, thus steerability with respect to ground is doubtful in any case.

Unfortunately, even with the spartan payload requirements and at the optimal season, sustained flight at strictly constant altitude above the Tropopause seems to be at the very limit of what is feasible today. Even if it could be achieved (according to the simulation) at higher latitudes in summer, the wind typically blows faster and renders such an undertaking doubtful. Definitely, more advanced large scale solar airplanes using regenerative fuel cells (as in NASA's Helios) for energy storage and/or allowing altitude change would have to be considered in order to achieve better perspectives.

2.5.2 Performance of a Small UAS

As a second case, it may be interesting to estimate the maximum performance of a given solar airplane configuration as a function of season and latitude. In the following, a model size rib wing airplane is analyzed (3 m wingspan, aspect ratio of 12, design battery mass of 2.35 kg, overall maximum mass of 4.2 kg). Again, a payload of 0.6 kg that consumes 4 W is given—which is considered the absolute minimum necessary to still perform some surveillance task, such as wildfire monitoring using a small thermal camera. The lower altitude limit is set to 700 m AMSL. Figure 2.8 shows the performance obtained with the design tool in terms of endurance, or, if sustained flight is possible, in terms of excess time. Notice the following results and interpretations:

- Sustained flight at constant (low) altitude is only possible in summer at medium to high latitudes.
- Allowing variable altitude increases the performance and extends the region of possible sustained flight to lower latitudes. However, the benefits are limited.
- The optimal battery weight changes drastically depending on the environmental conditions and also depending on whether or not altitude variation is allowed.

2.5.3 Low Altitude Record UAS

A final case study concerns record flying with a solar airplane at low altitude. Potentially, such an airplane could fly around the world or across the Atlantic ocean—at a certain latitude.

We chose the setting of low flight in order to limit regulatory issues (no interference with commercial air traffic operating in Instrument Flight Rules), as well as to avoid strong wind. Clearly this choice comes at the cost of having to rely upon the absence of significant amounts of clouds. We anticipate the region slightly poleward of the subtropic high pressure belts (around 38° North or South) to provide ideal conditions in the sense of sunny weather, little wind, and still reasonably long days in summer. The dominant

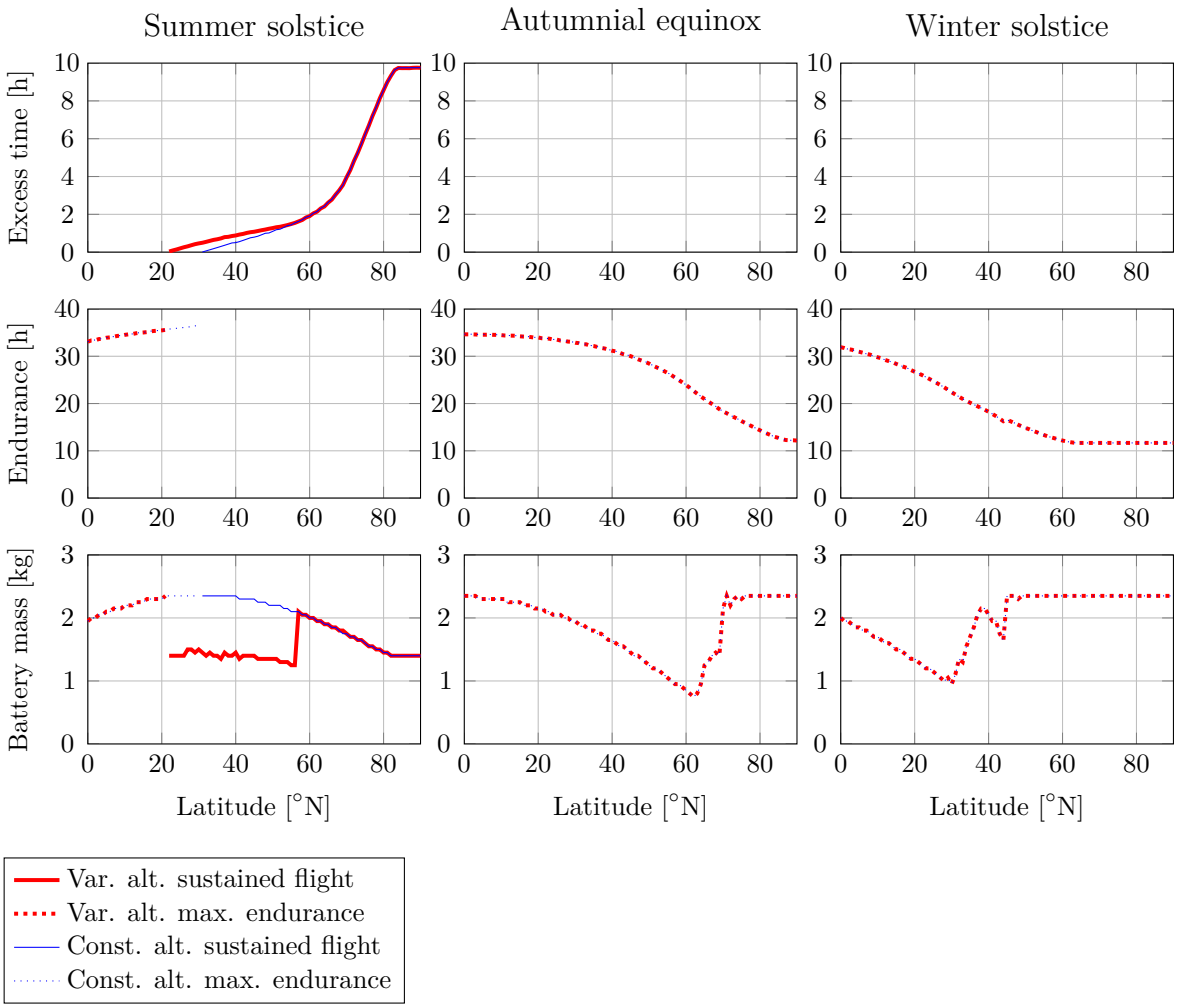


Figure 2.8: Performance and optimal battery mass of small solar airplane: the thick red lines stand for the case of allowed altitude variation; dashed lines indicate maximized endurance rather than excess time (solid lines).

wind direction will be west; therefore, the recommended flight direction is from west to east, which comes at the further advantage of reduced night times.

Figure 2.9 overviews the design space of such a record UAV as built using the rib wing structural variant. We plot the excess times for sustained flight, or endurance values as a function of wingspan and battery mass. The aspect ratio was fixed to $\Lambda = 18.5$ in this case, since this choice provides best performance in the interesting area of wingspan and battery mass. The corresponding nominal speeds and structural masses are also provided.

The following conclusions can be drawn from the figure: sustained flight at the indicated environmental conditions is assessed to be feasible by the design tool when choosing a wingspan above 3 m, as well as an appropriate battery mass. The margin (excess time) is maximized in the area of 8 to 14 meters of wingspan (quite flat optimum with the maximum value of 13.2 h at 11.5 m).

This analysis has determined the conceptual design of the AtlantikSolar airplane—aiming at performing the first-ever Atlantic crossing with an UAV solely relying on solar

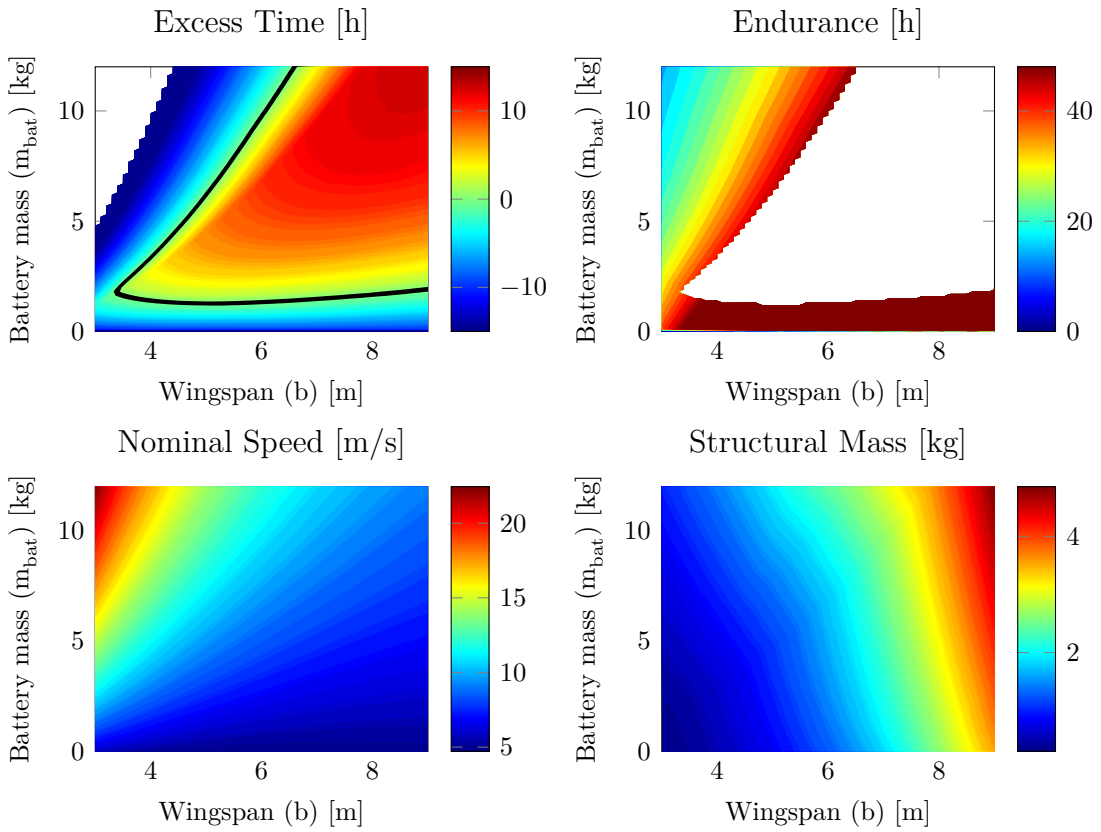


Figure 2.9: Record solar airplane design space (rib wing): for a fixed aspect ratio $\Lambda = 18.5$, the excess times for sustained flight, or endurance values are provided for varied wingspans and battery masses (first row). Corresponding nominal flight speeds and structural masses are given in the second row.

energy. In fact, a size considerably below the estimated optimum was selected, in order to limit complexity and cost of a single airplane, and still allow for easy deployment by hand-launch.

Concretely, the wingspan was fixed at 5.65 m and a battery mass was chosen as 2.9kg. With this concept at hand, we furthermore assessed the feasibility of the Atlantic crossing as a function of latitude and season. Respective results are provided in Figure 2.10. Judging from the plot, the Atlantic crossing would be feasible at the chosen latitude

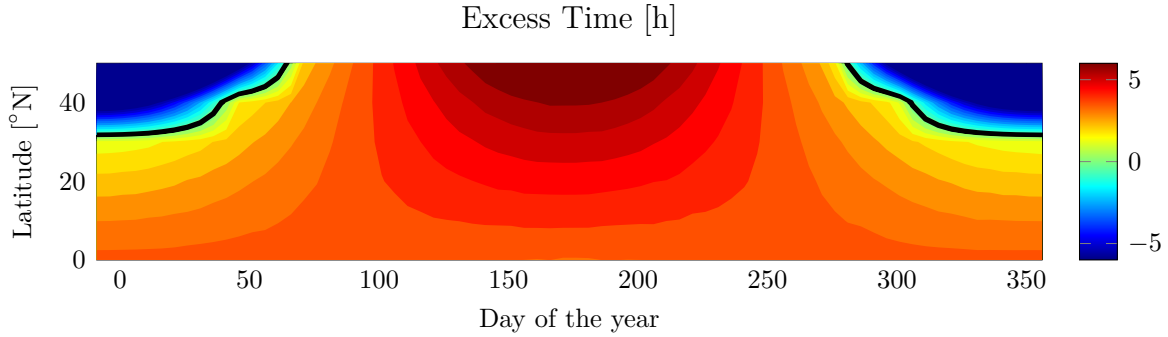


Figure 2.10: Atlantic crossing feasibility as a function of latitude and season.

(around 38° N) during an extended period around the optimal date, i.e. June 21st, where the margin amounts to 5 hours. Note that again, the margins calculated mark somewhat optimistic bounds of what is achievable with the current technology.

2.5.4 Comparison to Existing Airplanes

The described structural sizing and weight estimation is compared to some examples, in order to verify whether the values of the tool are realistic. Both the estimates obtained with the statistical model by Noth and the new model described in this chapter are listed in Table 2.2 for comparison with each-other and with the real structure masses.

Example	Wingspan	A. ratio	Noth	New	Reality	Remark
Unmanned Solar Airplanes: Rib Wing						
Sky-Sailor	3.2 m	12.7	875 g	509 g	700 g	batt. centered
SunSailor	4.2 m	13.13	2016 g	849 g	1700 g	batt. centered
AtlantikSolar	5.65 m	18.5	4638 g	1317 g	1800 g	batt. distr.
Zephyr	18 m	11.6	189 kg	51 kg	≈20 kg	batt. distr.
Unmanned Solar Airplane: Shell Wing (includes solar module)						
senseSoar	3.0 m	12.0	726 g	1300 g	1700 g	batt. distr.
Manned Solar Airplanes: Rib Wing (Pilot Centered)						
Goss. Penguin	21.6 m	13.86	365 kg	163 kg	31 kg	
Gliders: Shell Wing (Water Ballast Distributed, Pilot Centered)						
DG808	18 m	21	163 kg	127 kg	334 kg	
Ventus 2	18 m	23	159 kg	135 kg	230 kg	

Table 2.2: Different structure mass estimation for some examples.

The examples illustrate again what the tool was intended for: it gives a baseline for ultra-lightweight airplanes with a wingspan not much over 10 m. Not surprisingly, the two gliders are considerably heavier than the estimates, because they are designed for higher loads, easy handling and not with an extremely weight-optimized structure. The Gossamer Penguin mass, however, is highly overestimated: it is certainly not designed for higher load factors and large aileron deflection at high speed. The same supposedly applies in parts to Zephyr. Nevertheless, the estimates of both the Zephyr as well as the Gossamer Penguin structure mass are considerably closer to the real ones compared to the statistical model.

Most importantly, the suggested method models the three solar airplanes built at ASL to a high accuracy—and concerning the two newest ones, senseSoar and AtlantikSolar, with a much higher accuracy than the purely statistical method that neglects the influence of battery mass and other payload. In particular, the AtlantikSolar mass is dramatically overestimated by the statistical model (by a factor of 2.6), whereas the proposed tool slightly underestimates it (by 27 %)—again, the tool output has to be regarded as a lower bound of what can be achieved with the current technology.

2.6 Discussion

A multi-disciplinary optimization tool for solar airplane conceptual design was presented. One key improvement over existing approaches is the structural module which calculates a feasible simplified structure in order to predict its weight rather than relying on statistical data. Either a rib and a shell wing concept may be selected. Moreover, flight with variable altitude profile is investigated, which may enable better performance.

The simulation results show that sustained flight at constant altitude with minimal payload can only be performed above the Tropopause at medium to high latitudes in summer and at speeds that are lower than typical wind speeds. The optimal wingspan for such missions are found to be in the range of 13 m. When analyzing an RC-model size solar airplane with a minimal payload, the simulations show that it could fly sustained in the lower atmosphere at a large range of latitudes—that is slightly extended, if altitude variation is allowed. As a final case, the design space of a low-flying record airplane is analyzed: in essence, the tool predicts feasible sustained flight at a latitude around 38°, for wingspans above 3 meters, with an optimal energy margin in the area of 8 to 14 meters of wingspan. The latter two cases have determined the concept of the two most recent ASL solar airplanes senseSoar and AtlantikSolar, as described in the following Chapter 3. The structure masses estimated are in significantly better agreement with the real prototypes compared to the statistical approach.

As part of future work, the integration of more advanced structure concepts valid for larger wingspans together with a less conservative approach of imposing load cases would be of high interest: existing examples of ultra-lightweight solar airplanes (such as Solar Impulse or the Gossamer Penguin) show that structural masses can be lower than the presented prediction when scaling up.

Chapter 3
Solar Airplane Platforms

UNIQUE applications are enabled by solar airplanes owing to their autonomy in terms of flight time, as well as concerning navigation without operator intervention even close to the ground. Two prototypes were built at ASL during the past five years that we will overview in the following. On the one hand, the *senseSoar* airplane was intended to act as the robotics research platform not targeting eternal flight, but equipped with advanced sensing and processing capabilities. On the other hand, the *AtlantikSolar* prototype aims at extended record flying, while it may still be outfitted with a generic sensing and processing pod. This Chapter creates the link between conceptual design, flight testing and running vision supported localization and mapping algorithms tailored to robust airplane operation close to the terrain.

3.1 Introduction

With significant advances in solar cells, battery, sensing and processing technology that the past 40 years have experienced, a plethora of different solar airplanes has been realized, targeting both sole record flying as well as actual missions. Chapter 2 overviews some of the most prominent examples. An extensive treatment of solar airplane history can be found in [Noth 2006].

During the course of five years, two different solar airplane concepts as shown in Figure 3.1 were realized at ASL, both of which were inspired by the experimental proof of sustained flight with the small-scale UAV *Sky-Sailor* [Noth 2008] in 2008.

Autonomous solar airplanes are captivating owing to their potential of long-lasting flight in order to perform a particular task; while Sky-Sailor was targeting record flying, time had come to advance autonomous navigation capabilities close to the ground that would allow flying an actual mission, such as mapping, surveillance, or in the context of Search and Rescue (SaR). To this end, the *senseSoar* solar airplane was developed in collaboration with Leichtwerk AG[1], equipped with a suite of sensors and processors, mainly targeting vision supported localization and mapping rather than eternal flight. With ease of use and robustness to harsh weather conditions as complementary design criteria, a shell structure with fiber reinforced plastic and a relatively small size of 3.1 meters of wing span was chosen.

Again aiming at record flying, the development of *AtlantikSolar* started a little later, with the goal of extending the boundaries of sustained flight: eventually, as the name

[1]Leichtwerk AG, Lilienthalplatz 5, Braunschweig, Germany. See www.leichtwerk.de.



Figure 3.1: senseSoar and AtlantikSolar airplanes: senseSoar, featuring a wingspan of 3 meters is built completely from fibre reinforced plastics. AtlantikSolar, however, aiming at record flying, employs a rib and spar wing structure of 5.65 meters in span.

suggests, the goal will be to cross the Atlantic Ocean from West to East during summer and at low altitudes. To this end, a substantially larger wingspan than for senseSoar was chosen, namely 5.65 meters, as well as a traditional rib and spar wing construction. The development and testing of AtlantikSolar lies outside the scope of this thesis[2]; nevertheless, we will briefly overview its main characteristics for two reasons: first of all, it constitutes the realization of a concept generated with the conceptual design tool presented in Chapter 2, so the interested reader can make the connection from the concept stage to the actual implementation. Second, we developed a modular sensing and processing unit which we also attach to AtlantikSolar, in order to perform localization and mapping tests.

3.1.1 Solar Airplane Design Contributions and Outline

The contributions presented in the following mainly span engineering achievements that enable test campaigns which are novel to the field.

- We present two solar airplane prototypes that were built and successfully operated. All their components are custom-tailored and optimized towards a specific application scenario, thus validating the suggested conceptual design scheme presented in Chapter 2.

- We provide a sensing and processing infrastructure tightly integrated with the senseSoar airplane and as a modular pod that can be attached to the AtlantikSolar airplane: the systems described here enable long-endurance flight tests including

[2]The AtlantikSolar project is led by Philipp Oettershagen at ASL and supported by many partners and individuals, e.g. Dieter Siebenmann and Rainer Lotz responsible for the airframe design and manufacturing. See www.atlantiksolar.ethz.ch.

localization and mapping supported by cameras as described in Chapters 5 and 7. To the best of our knowledge, the combination of endurance flights with a small and easy-to-use airplane and such sensing and processing capabilities is unique.

This chapter is organized as follows: we first introduce the senseSoar airplane, including the airframe, power train as well as sensing and processing electronics in Section 3.2. Second, we overview the AtlantikSolar airplane in 3.3 with a focus on the modular sensing and processing unit.

3.2 senseSoar

Here we provide a short overview of the senseSoar design process and components. ASL was responsible for the elaboration of the overall concept and for the design of the power train as well as for the realization of sensing and processing electronics. Leichtwerk designed and analyzed the airframe in terms of aerodynamics, flight mechanics and structure.

3.2.1 System Overview

Figure 3.2 shows the configuration and sizes of the final prototype exported from our Computer Aided Design (CAD) software. The layout with two propellers and an inverted V-tail was chosen for the following reasons:

- Two motors for redundancy of thrust: one propulsion unit will deliver enough power to stay airborne and perform a landing (while the yaw moment is compensated by the tail). Of course the setup with two motors rather than one comes at the expense of some efficiency.
- Redundancy of steering moments: the setup with two motors and a V-tail allows for failure of any of the control surfaces: differential thrust can be used to generate a yaw moment (with low bandwidth).
- High currents, vibrations, as well as obstructed field of view by the propulsion units are spatially separated from the central sensing unit.
- Foldable propellers allow for efficient gliding mode and landing without dedicated mechanisms such as gear.
- The inverted V-tail accommodates a part of the solar module without the potential of shadows cast by one tail on the other.
- For considerations of space savings as well as load distribution (in order to save structural weight), the batteries are integrated inside the main wing.
- For easy transportation, the airplane can be disassembled into a central wing section with sensor pod, two fuselages with tail and propulsion unit, as well as two outer wings.

Figure 3.3 shows a rendered highlevel overview of the senseSoar components, including the accommodation of the battery packs inside the wings.

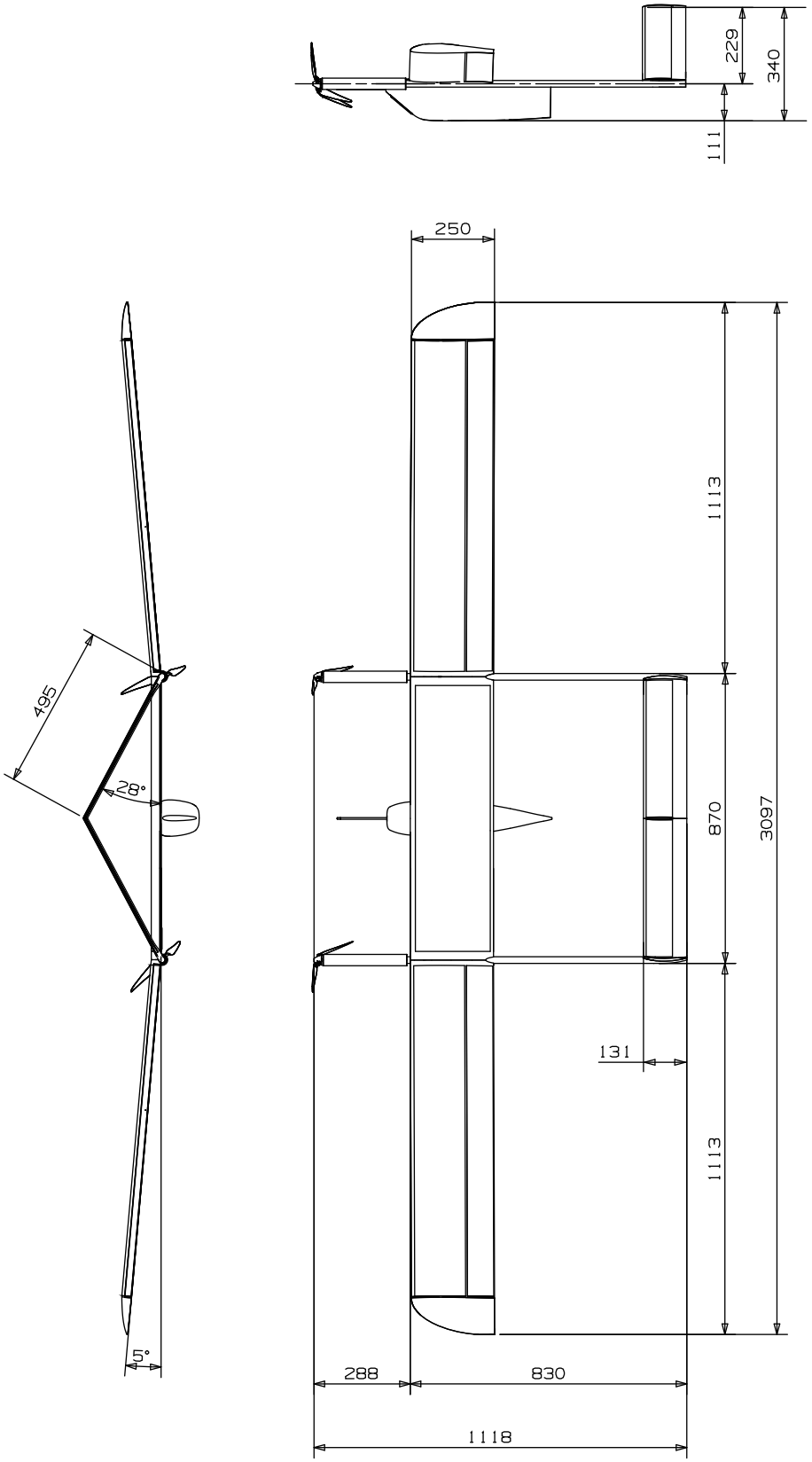


Figure 3.2: senseSoar views from left, behind and top (CAD) with the main dimensions in mm.

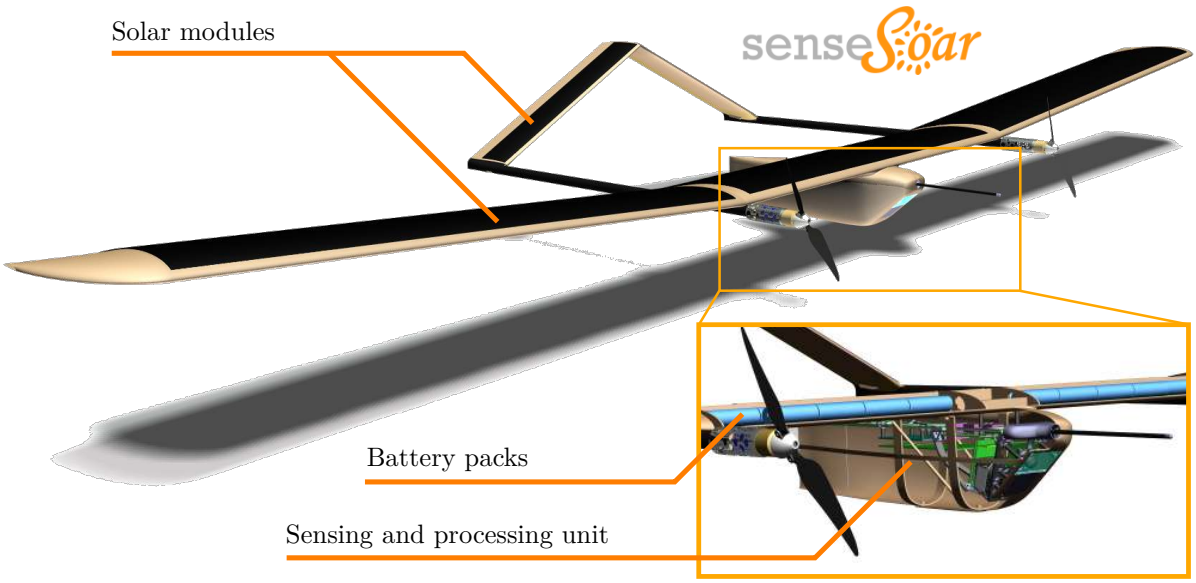
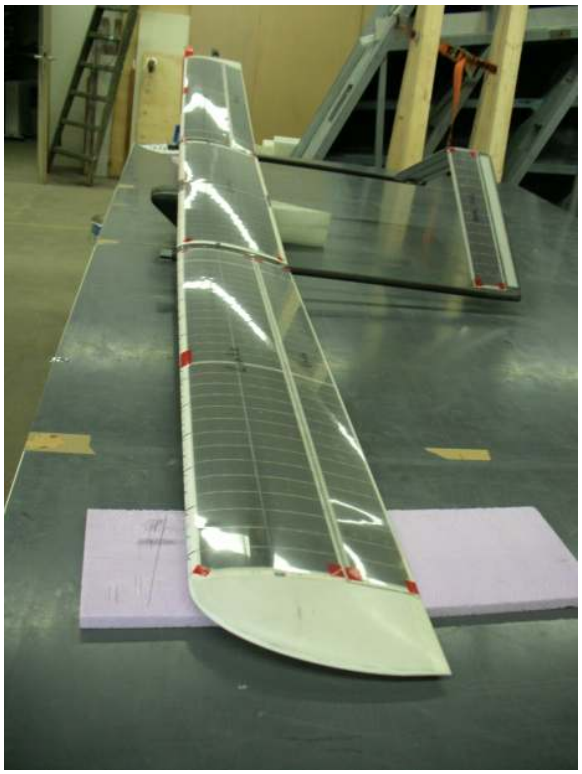


Figure 3.3: senseSoar components highlevel overview.

The decision was taken to manufacture two versions, both for redundancy and in order to test manufacturing processes: therefore a first version was built with dummy solar cells for testing. Figure 3.4 shows this first version as well as the actual solar airplane.



(a) First senseSoar version with dummy solar cells.



(b) senseSoar airplane with solar module.

Figure 3.4: The two solar airplane prototypes built by Leichtwerk AG: a first version employed dummy solar cells, in order to test the manufacturing processes and to have a prototype ready for testing early.

The main characteristics of the non-solar version are listed in Table 3.1.

Characteristic	Symbol	Value	Unit
Wing span	b	3.1	m
Wing area	A	0.78	m^2
Aspect ratio	Λ	12.3	-
Overall mass	m	5	kg
Battery mass	m_{batt}	2.0 (new 1.75)	kg
Stall indicated airspeed	v_{min}	9	m/s
Nominal indicated airspeed	v_{min}	10.4	m/s
Maximum indicated airspeed	v_d	30	m/s

Table 3.1: senseSoar characteristics.

Further analysis of static aerodynamics and dynamic properties of senseSoar that have influenced the design process will be introduced in Chapter 4.

3.2.2 Airframe

The choice was made for a fully fibre reinforced structure, laminated in negatives, including the main wing and tail with embedded solar modules—for which Leichtwerk AG is a leading expert in the field. While the concept results in heavier structures than equivalent traditional rib and spar concepts, this method still offers some major advantages. First of all, the structure will be more *robust*, owing to its outer hard shell. Second, there exist standard procedures of calculating such a structure (as employed commonly in manned glider design)—upon which a potential future certification could be based. Last but not least, once the negatives are available, the reproduction of new senseSoar instances is supposed to be comparably straight-forward.

Leichtwerk has made a considerable effort to achieve the characteristics in terms of aerodynamic performance and flight dynamics as desired by the specific application. In collaboration with the aerodynamicist Dr. Martin Hepperle, the airfoil MH139F was developed and used for the senseSoar main wing.

Figure 3.5 shows the wing shell halves consisting of an outer glass fiber layer, foam and diagonal carbon fiber rovings on the inner side. Note that the solar module is laid into the negatives first and acts as the outer shell of the wing top side.

3.2.3 Power Train

The power train consists of the solar module, a Maximum Power Point Tracker (MPPT), a Lithium Ion battery pack as well as of the propulsion unit. Maximum efficiency of the power train is again of core interest—which includes minimizing its weight.

Solar cells for small UAVs need to be both bendable at rather low radii, as well as deliver maximum weight specific power: therefore, Azur Space S32 thin conventional silocon cells[3] were selected, equivalent to the Sky-Sailor cells. The cells are sized only 74 × 31.9 mm, which allows for some freedom on the layout that needs to fit with the wing geometry. They furthermore feature a specific mass of 32 mg/cm^2 at an efficiency around

[3]Datasheet available at www.azurspace.com as of May 2014.



(a) Shells lying in the negatives at an early stage (non-solar version).



(b) Shells in the negatives before closing (solar prototype).

Figure 3.5: Impressions from the lamination process at Leichtwerk in Braunschweig: senseSoar wing shells lying in the negatives.

17 to 19 % in the lower atmosphere, which corresponds to a very high power-to-mass ratio (today's highest efficiency series production GaAs triple junction cells are significantly lower). The voltage ranges from 0.528 V (maximum power) to 0.628 V (open circuit).

When designing the layout of the solar module, the choice was made to use a relatively low battery voltage of nominally 12 V (three Lithium Ion cells in series), in order to easily connect various electronics subsystems. This decision obviously necessitates larger cable cross sections and connectors, which implies higher weight. Since boost-type of MPPTs were foreseen, the solar module voltage must not exceed the battery voltage, which has led to the layout depicted in Figure 3.6. In essence, three modules are placed on the wing and the tail. While all three modules connect 14 cells in series, two modules consist of four such strings in parallel, and one module of only three—i.e. 2 × 14S4P and 1 × 14S3P. The modules were manufactured by Gochermann Solar Technology. [4]. They are connected to a central power board that accommodates the MPPTs (developed in-house, but currently being replaced with a version based on an off-the-shelf chip).

Four battery packs are placed between spar and leading edge inside the main wing. We use the Panasonic NCR18650 and NCR18650B Lithium Ion cells[5], since they offer the best commercially available gravimetric energy density. In numbers, the cells operate at 3.7 V nominal at a capacity of 2900 mA and 3150 mA—while featuring a weight of

[4] See www.gochermann.com, accessed May 2014.

[5] Datasheets available at industrial.panasonic.com, accessed May 2014.

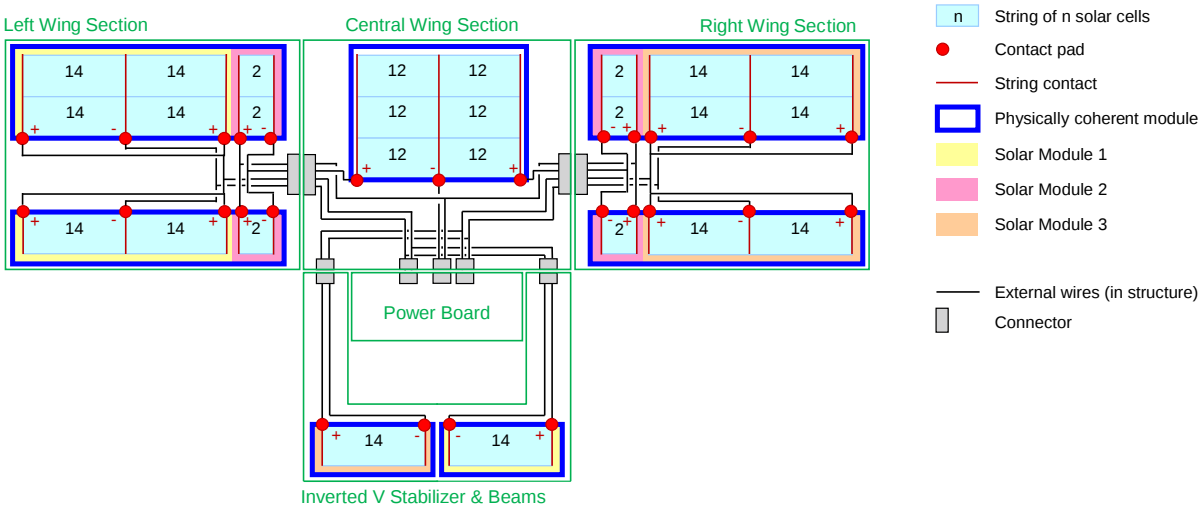


Figure 3.6: senseSoar solar modules layout: three modules with 14 solar cells in series each are distributed on the wing and tail.

45.5 g and 46.5 g, respectively. The middle wing accommodates two packs of type 3S3P and the outer wings each contain a 3S4P pack (3S3P in a newer version employing the NCR18650B cells).

The final propulsion group design uses a Scorpion HK 2216/16 Brush-Less DC (BLDC) outrunner with a Reisenauer Micro Edition 5:1 planetary gear reduction[6]. The motor parameters are given as speed constant of 2608 rpm/V, no-load current of 1.19 A (at 10 V) and a resistance of 0.048 Ω. A 16×13 inch foldable two-blade carbon propeller is used. Figure 3.7 shows the setup, where the motors, gearboxes as well as motor controllers are accommodated inside carbon tubes extending from the wing and fuselages.



Figure 3.7: senseSoar propulsion units: motors with gearbox and controllers located in carbon tubes drive foldable carbon propellers.

[6]Datasheets for motor and gearbox available at www.reisenauer.de, accessed May 2014.

3.2.4 Sensing and Processing Electronics

The components accommodated inside the central pod serve several purposes. First of all, the power board contains MPPTs and several DC-DC converters, in order to power all systems. Second, a basic autopilot system is based on the open hardware and software project px4[7]. The px4 microcontroller board uses an Inertial Measurement Unit (IMU), static and dynamic pressure sensing via the pitot-tube, as well as GPS and a three-axis magnetometer in order to perform state estimation as outlined in Chapter 5 and runs the autopilot that outputs servo positions and motor setpoints. Third, the highlevel computer was intended to be used for advanced state estimation which includes imagery from both a visible light color camera and a FLIR Tau 320 thermal camera; application-related tasks, such as people detection and mapping were also foreseen to be deployed on this processing unit. Figure 3.8 overviews the components of the pod. As a

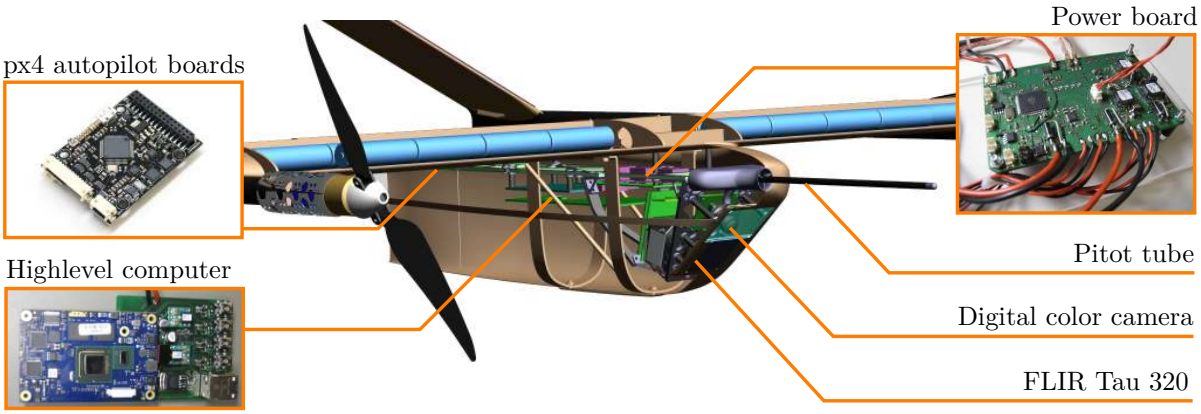
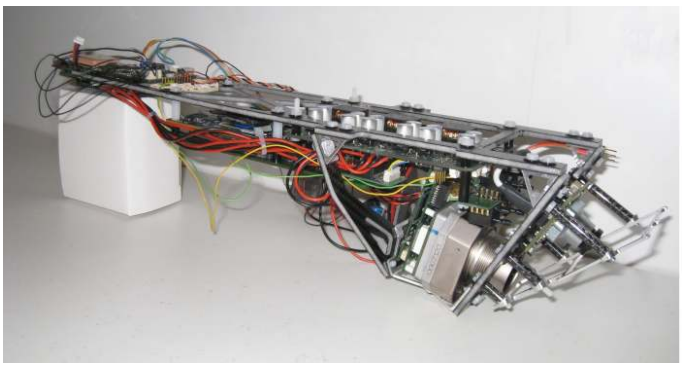


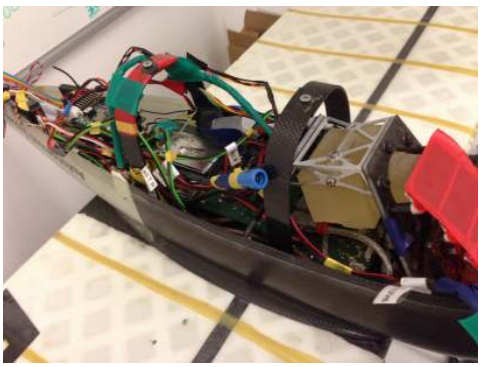
Figure 3.8: senseSoar pod overview.

highlevel processor board, an Intel® Atom™ Z530 (1× 1.6 GHz) based card was used on top of a minimalistic base board developed in-house.

Figure 3.9 shows the actual assembly separated and integrated into the airplane.



(a) senseSoar pod assembly.



(b) senseSoar pod airplane integration.

Figure 3.9: senseSoar pod assembly and integration: for first test flights, the FLIR thermal camera was replaced by a dummy.

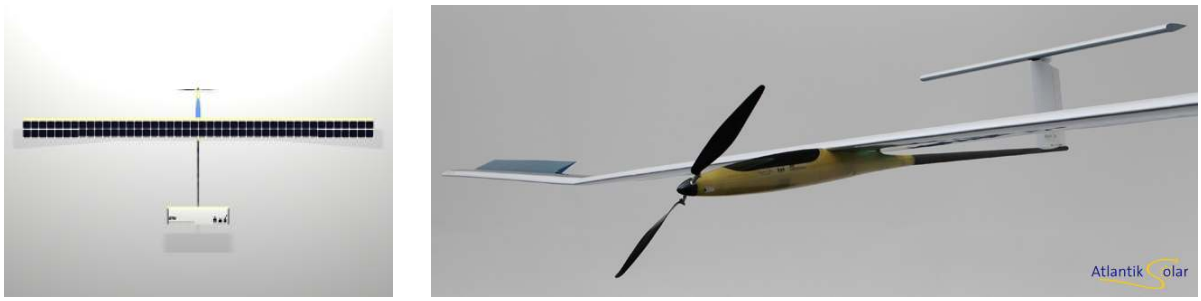
[7] See pixhawk.org, accessed May 2014.

3.3 AtlantikSolar

The AtlantikSolar project clearly targets record flying, with the ultimate goal of crossing the Atlantic Ocean at low altitudes in Summer 2015. Chapter 2 describes the approach followed to determine the AtlantikSolar conceptual design; here, we quickly overview the actual platform and focus on the modular sensing and processing unit that is aimed at extending the applicability of the airplane to mapping, surveillance and SaR tasks.

3.3.1 System Overview

A top view CAD rendering and a photograph of the actual airplane are provided in Figure 3.10. A conventional configuration with a single fuselage and single propulsion



(a) AtlantikSolar top view CAD rendering.

(b) AtlantikSolar airplane close-up.

Figure 3.10: The AtlantikSolar airplane. A conventional configuration was chosen that employs one propulsion unit attached to one central fuselage. The wing is built as a rib and spar structure.

unit was chosen, in order to optimize for weight and power consumption. The wing is built using a traditional rib and spar concept: the spar is implemented as a carbon tube that also accommodates the batteries such that the load is distributed along the wing.

Some of the core characteristics in terms of size, mass and speeds of the AtlantikSolar UAV are listed in Table 3.2

Characteristic	Symbol	Value	Unit
Wing span	b	5.65	m
Wing area	A	1.72	m^2
Aspect ratio	Λ	18.5	-
Overall mass	m	6.7	kg
Battery mass	m_{batt}	2.91	kg
Stall indicated airspeed	v_{min}	7.2	m/s
Nominal indicated airspeed	v_{min}	9.7	m/s
Maximum indicated airspeed	v_d	20	m/s
Maximum climb rate	$v_{c,max}$	2	m/s
Maximum sink rate	$v_{s,max}$	3	m/s

Table 3.2: AtlantikSolar characteristics.

3.3.2 Sensing and Processing Pod

A modular sensing and processing unit was designed that can be attached under the AtlantikSolar wing, in order to accomplish missions that reach beyond pure record flying. Figure 3.11 shows the assembled unit. The components setup has emerged from the

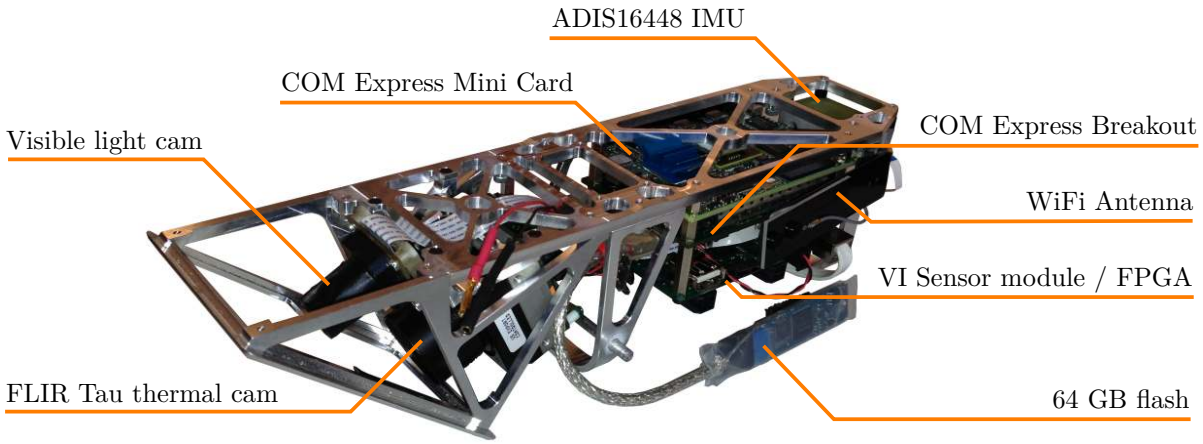


Figure 3.11: Modular sensing and processing unit that can be mounted under the AtlantikSolar wing.

senseSoar pod: in essence, a visible light monochrome WVGA (752 × 480) camera and a FLIR Tau 2 thermal camera with a resolution of 640 × 512 pixels[8] are interfaced using the ASL/Skybotix visual-inertial sensor[9] module (with FPGA) and ADIS16448 IMU as presented in [Nikolic et al. 2014] that is used also for the experiments in Chapter 5. The mechanical mount has been adapted to fit the particular needs of fixed-wing flight. The cameras are tilted in such a way that the Fields of View (FoV) cover the important area of the flight direction and 15° above, as well as down to the ground as far as possible. The vertical FoV of the visible light camera amounts to 90° with the current lens choice, and the FLIR provides 69°.

A GigE connection interfaces the visual-inertial unit with breakout board developed at ASL that accommodates any of Kontron's COM Express® mini pinout type 10 modules[10]. We have used as modules Intel® Atom™ D2550 (2× 1.86 GHz, 10 W) and N2800 (2× 1.86 GHz, 6.5 W), as well as the newer Atom™ E3845 (4× 1.91 GHz, 10 W) and Celeron® J1900 (4× 2.00 GHz, 10 W) boards—that allow for real-time processing of the algorithms outlined in 5 and 6.

The unit is connected to the px4 autopilot system of the airplane via a serial link that employs the MAVlink protocol. The setup will be ported to act as senseSoar's new sensing and processing infrastructure, and we furthermore are in the process of integrating it into our Techpod test airplane[11].

[8] Documentation is available at www.flir.com/cvs/cores, accessed May 2014. The 9 mm lens option was chosen.

[9] Specifications available at www.skybotix.com, accessed May 2014.

[10] See www.kontron.com/products/computeronmodules/com-express/com-express-mini, accessed May 2014.

[11] See hobbyuav.com, accessed May 2014.

3.4 Discussion

Two ASL solar airplanes with complementary characteristics were presented: senseSoar was designed to push the frontiers of navigation autonomy, while AtlantikSolar aims at achieving new records concerning endurance. The components of the senseSoar platform were overviewed in detail as an illustration of how conceptual design presented in Chapter 2 is transferred into an actual prototype tailored to operation at low altitude; the latter includes the ability for state estimation as described in 5 and validated in-flight as outlined in Chapter 7. Further tools for analysis of static and dynamic senseSoar properties as they were employed during the design process are summarized in the next Chapter 4.

The AtlantikSolar airplane was developed outside the scope of this thesis. However, we use it as an additional testing platform to which a modular sensing and processing unit can be attached, which we presented here in detail, and which will form the basis of the next iteration of the senseSoar pod.

Chapter 4
Airplane Dynamic Models

Modeling airplane characteristics plays a major role throughout the design phase: analysis of static aerodynamic as well as dynamic properties are used to assess load cases, and to ensure that application-specific requirements as well as handling qualities are met. Furthermore, an accurate simulation environment serves autopilot testing, pre-tuning but also operator training during and after the airplane development. Moreover, a simplified, parametric model can be directly used for model-based control, such as Linear Quadratic Regulators (LQR) with gain scheduling or Nonlinear Dynamic Inversion (NDI).

In the specific case of unmanned solar airplane design, modeling was of particular importance in order to assess the aerodynamic efficiency and to develop a suitable controller. Furthermore, knowledge of aerodynamic properties enables more robust and accurate state estimation including the estimation of wind, angle of attack and sideslip angle, as explained in detail in Section 5.4.

In the following, we present our 2.5D Nonlinear Lifting Line (NLL) aerodynamics analysis tool based on 2D airfoil properties. We then approximate this high-fidelity simulation with a parametric model chosen to match the special requirements of a slow-flying solar airplane. We apply the method to the senseSoar airplane as described in Chapter 3 and provide its main aerodynamic characteristics.

Finally, the full nonlinear dynamic model is provided, along with an analysis of the linearized longitudinal and lateral dynamics of the aforementioned platform.

4.1 Introduction

Modeling the 3D flow field along with the resulting forces and moments around an airplane is a complex undertaking: Computational Fluid Dynamics software (CFD) solves for the underlying Navier-Stokes partial differential equation using Finite Elements (FE). This brute-force approach yields the most accurate results that may be obtained with computational methods; the process, however, requires expert knowledge in using such software (setting parameters, meshing) and solving for just one particular control configuration and state takes comparably long.

As more tractable yet still reasonably accurate alternative, literature suggests 2.5D methods. Assuming an inviscid and incompressible fluid, the flow may be modeled with potential field theory (for an extensive treatment see e.g. [McCormick 1979]), where the velocity vector field is modeled as the gradient of a scalar function. This concept allows

for insertion of singularities into a free stream, such as sources, sinks and vortices, in order to capture the major effects of the 3D velocity field. In particular, analyzing lift and drag polars for a finite wing rather than just for its airfoil reveals less lift increase per angle of attack increment, less maximum lift, and higher drag at raised angles of attack. These observations are related to the concept of *induced flow*, explained in the following.

The *Lifting Line* model uses horseshoe vortices to model induced downward flow deflection, which is a direct consequence of lift and explains the aforementioned effects of a finite wing as opposed to a 2D flow across an airfoil. Figure 4.1 shows a first approximation using a single vortex—conceptually illustrating the flow characteristics around a simplified wing. The vortex system consists of the bound vortex and tip vortices;

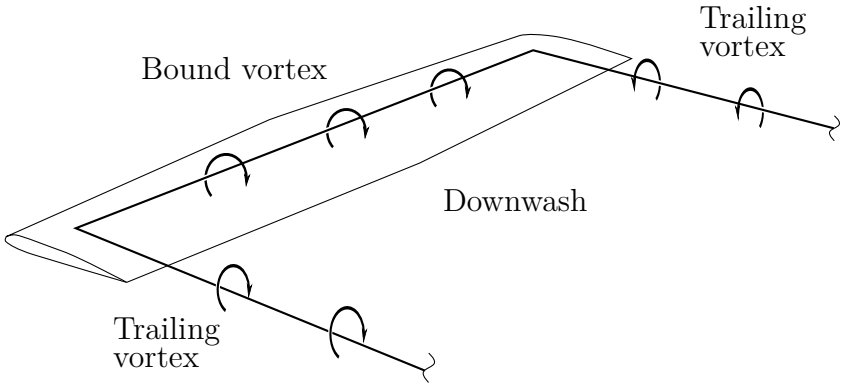


Figure 4.1: Simplified representation of the wing vortex system: as a consequence of lift, a bound vortex is formed along with trailing wingtip vortices. The vortex system induces downwash, present mostly behind the wing but to some extent already developped at the wing.

note that the vortex will in theory have to be closed to form a ring by considering a "starting vortex". But in practice, i.e. in presence of friction, the vortices will decay over time. Figure 4.2 illustrates the existence of tip vortices trailing an airplane wing. The



Figure 4.2: Wake Vortex Study by NASA at Wallops Island: the tip vortices are visualized using colored smoke rising from the ground.

vortices induce a downwash area behind the wing; nevertheless, the trailing vortices will also induce some downward flow at the wing.

For more accurate flow field calculation, the Lifting Line method typically uses a finer discretization along the wing into many horseshoe vortex elements [McCormick 1979]; the *Nonlinear Lifting Line* method, which we adopt, considers full 2D airfoil polars, rather than just approximating them as linear relationships of Angle of Attack (AoA) and lift coefficient, as first suggested in [Sivells and Neely 1947]. A further approach for increased accuracy is known as *Vortex Lattice Method* (or lifting surface method described e.g. in [McCormick 1979]): here, vortices are not only distributed in spanwise direction, but as a 2D array on the cambered surface. Any of these methods may be furthermore be complemented with fuselage body modeling using doublet or source/sink distributions combined with a respective boundary layer approximation.

Note that tools for 2D airfoil flow computation such as JavaFoil[1], xfoil[2], or XFLR[3] are available free of charge and thus widely used by the RC-modeling community. Furthermore, free 2.5D tools such as AVL[4] and XFLR implement some of the methods as described above, and are thus also very popular e.g. for RC model development.

4.1.1 Outline and Contributions of the Modeling Methods

The methods presented in the following have been extensively discussed in literature. We describe our complete toolchain that allows for analysis of electrically driven airplane designs both in terms of static as well as dynamic properties; it enables performing real-time simulations, and obtaining a fully parametric dynamic model that can be used for both accurate state estimation and model-based autopilot synthesis. Specifically, the approach consists of the following elements:

- We describe in detail the 2.5D aerodynamics simulation based on a Nonlinear Lifting Line technique interfaced such that the user simply has to specify the airplane geometry layout and airfoils employed. The method and C++ implementation allows for real-time simulations.
- A fully parametric version of airplane dynamics is presented which may be automatically derived from the 2.5D simulation, including the model for electrically driven propellers.
- We describe the dynamics analysis procedure in terms of longitudinal and lateral linearized dynamics.
- We analyze the senseSoar solar airplane concerning static performance as characterized by polars. Dynamic properties are assessed in the form modes associated with the pole locations of the linearized plant.

The remainder of this chapter is structured as follows: in Section 4.2, we first overview the notation and definitions associated with aerodynamics and flight mechanics employed. Section 4.3 presents the Nonlinear Lifting Line aerodynamics simulation, which is then followed by the description of nonlinear airplane dynamics in the form of a fully parametric model in Section 4.4, including also propulsion modeling and linearization. Section 4.5 finally applies the proposed methods to senseSoar.

[1] Available free of charge at www.mh-aerotools.de/airfoils/javafoil.htm as of March 2014.
[2] Available free of charge at web.mit.edu/drela/Public/web/xfoil as of March 2014.
[3] Available free of charge at www.xflr5.com as of March 2014.
[4] Available free of charge at web.mit.edu/drela/Public/web/avl, accessed March 2014.

4.2 Notation and Definitions

In the following, we introduce the foundations upon which our modeling approach. The notations and definitions largely follow aerodynamics and flight mechanics standards in order to be consistent with [Etkin 1972], [McCormick 1979] and [Ducard 2009].

4.2.1 2D Flow Around Airfoils

Historically and for practical reasons, the aerodynamic force is split into a component perpendicular to the inflow direction called lift, and a second one parallel to the inflow called drag. We write 2D lift, drag and moment as infinitesimal quantities dL, dD, and dM, respectively, as opposed to L, D, and M designating physical forces of a whole airplane. Figure 4.3 visualizes these quantities. Furthermore, the angle of attack α is defined as the angle between inflow direction and the chord line of length c connecting airfoil leading edge and trailing edge. Note that force and moment are reduced to the point at $0.25c$, i.e. one quarter of the chord behind the leading edge.

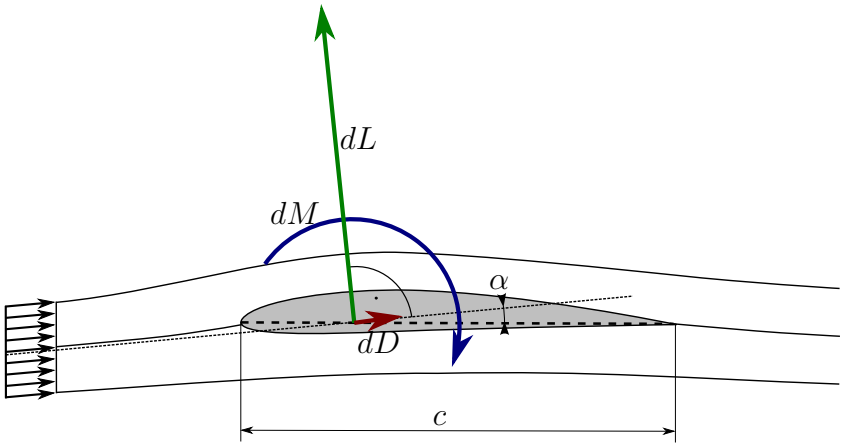


Figure 4.3: Decomposition of the aerodynamic force by a 2D flow around an airfoil: lift dL denotes the component perpendicular to the inflow, and drag dD the one parallel to it.

Dimension analysis suggests the formulation of aerodynamic forces and moments in terms of physical quantities along with dimensionless section lift, drag, and moment coefficients c_l, c_d, and c_m:

$$dL = \frac{1}{2}\rho V_t^2 c_l c dy, \tag{4.1}$$

$$dD = \frac{1}{2}\rho V_t^2 c_d c dy, \tag{4.2}$$

$$dM = \frac{1}{2}\rho V_t^2 c_m c^2 dy, \tag{4.3}$$

where V_t stands for the inflow speed, ρ for the air density and dy denotes an infinitesimal length element perpendicular to the 2D flow (which can be interpreted as a length element into span-wise direction of an infinitely long wing).

These coefficients largely depend on the angle of attack α; but furthermore, the Reynolds and Mach numbers significantly influence them as well. The angle of attack dependencies are typically given in the form of section lift, drag and moment polars, an

example of which is provided in Figure 4.4. Note that the drag component is originating both from viscous skin friction as well as form drag, caused by an asymmetric pressure distribution due to boundary layer development and separation.

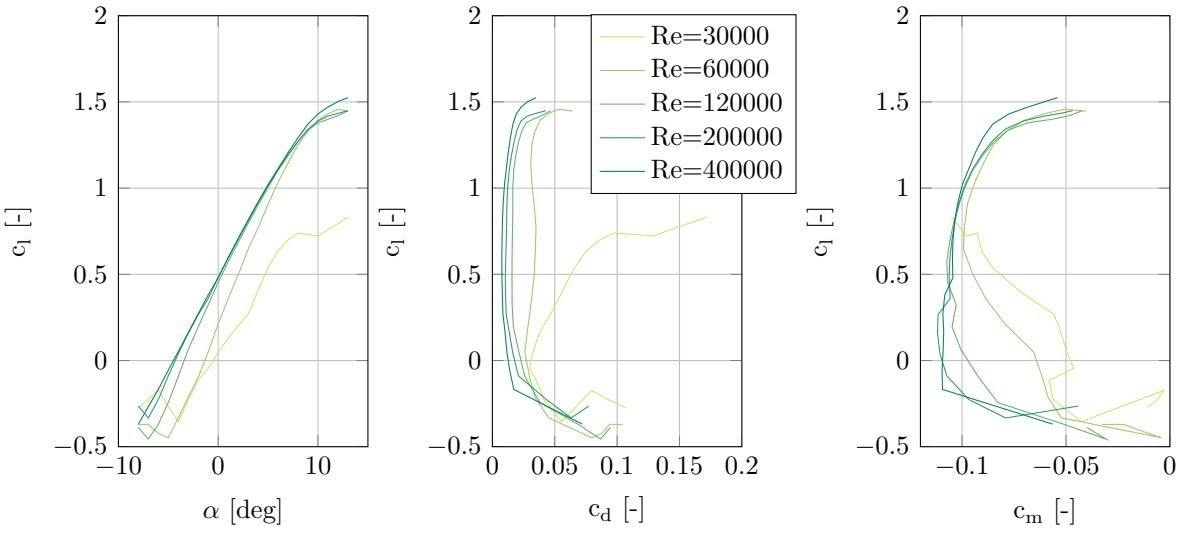


Figure 4.4: Lift, drag and moment polars for various Reynolds numbers. The polars were calculated using xfoil on the MH139F low-speed airfoil designed by Dr. M. Hepperle for senseSoar and other ASL solar airplanes.

The lift curve shows its characteristic linear increase with increasing α for small angles of attack. The maximum and minimum lift values beyond which stall is entered are visible in the lift polar. Notice that the aerodynamic performance c_l/c_d of the airfoil generally decreases with smaller Reynolds numbers as expected. The choice of reference point at $0.25c$ typically leads to a mostly constant moment coefficient when varying α, c_l, respectively, as can be seen in the Figure 4.4 as well.

4.2.2 Forces and Moments Acting on an Airplane

Consider Figure 4.5 for the introduction of airplane geometry definitions and main forces. Forces and moments are reduced to the airplane Center of Gravity (CoG). Note that the Angle of Attack (AoA) α is defined as the angle between the true airspeed vector $\mathbf{v}_t$ projected into the body x-z-plane and the x-axis. β denotes the sideslip angle, causing a typically unwanted sideslip force Y. L and D denote lift and drag, W stands for the weight force and T for thrust, which may act into a direction different from x (at a thrust angle ϵ_T). We furthermore write the aerodynamic moment vector as ${}_B\tau_A = [L_A, M_A, N_A]^T$. Also note the introduction of the main control surfaces that are designed to mainly influence the aerodynamic moment: with ailerons, elevator, and rudder, the roll (L_A), pitch (M_A) and yaw (N_A) moments are controlled. The indicated flaps, if available, are used for increasing lift for take-off and landing, in order to achieve a slower minimum speed.

Aerodynamic Forces and Moments

A large part of the method described below is concerned with using the 2D polar information, in order to calculate forces and moments for the overall airplane: they may

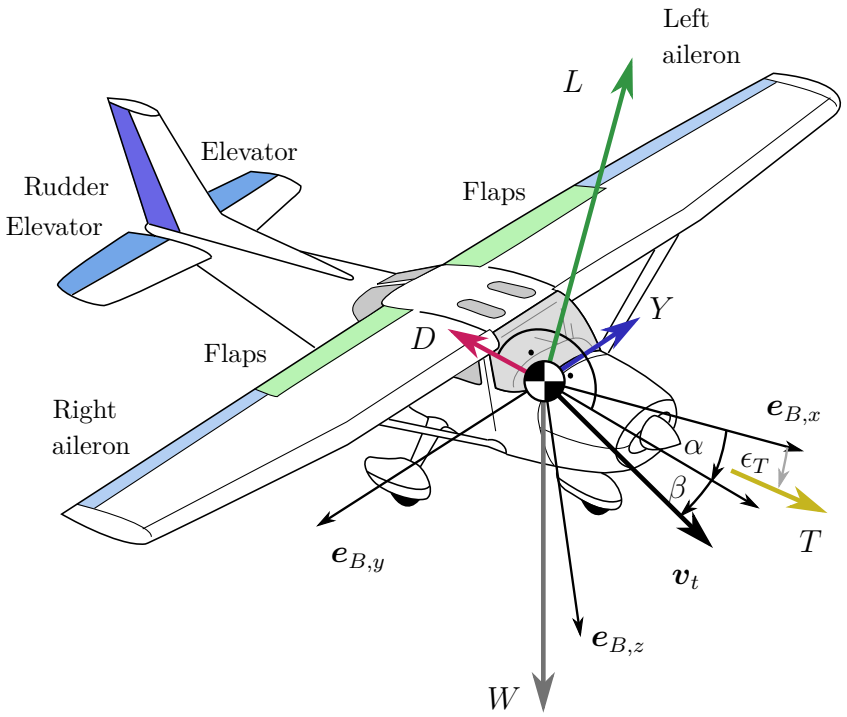


Figure 4.5: Geometric definitions and main forces acting on the airplane (general case, not in equilibrium).

again be written with dimension-less coefficients, such as:

$$L = \frac{1}{2}\rho V_t^2 c_L A, \tag{4.4}$$

$$D = \frac{1}{2}\rho V_t^2 c_D A, \tag{4.5}$$

$$M_A = \frac{1}{2}\rho V_t^2 c_M \bar{c} A, \tag{4.6}$$

with the wing area A, the mean chord length $\bar{c}$, and the true airspeed $V_t = \|\mathbf{v}_t\|$. The moments L_A and N_A are made dimension-less with the wingspan b rather than the chord length. Note the capital coefficient indices.

4.3 Nonlinear Lifting Line

In the following, we present one example of how to numerically approximate the lift and drag distribution of a wing including induced drag.

4.3.1 Discretized Representation of the Lifting Surfaces

The lifting line method is a 2.5D approach in which the induced flow is viewed as generated by several discrete horseshoe vortices rather than just one as introduced qualitatively above. Figure 4.6 introduces the geometry and variables involved. Note that the method only provides reliable results, if the assumption holds that spanwise flow is negligible; in particular, spanwise variation of parameters such as chord length and twist is supposed to be rather small.

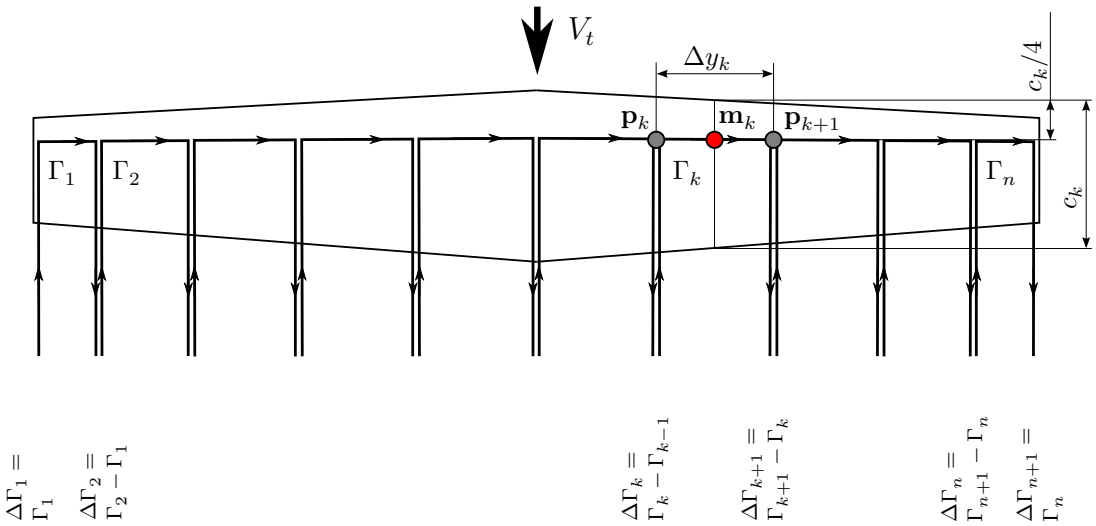


Figure 4.6: k horse shoe vortices with circulation Γ_k placed on the wing to model the induced flow. The lifting line is imagined through the quarter-chord ($c_k/4$) locations. Vortex threads with strength $\Delta\Gamma_k$ are leaving the wing at the points $\mathbf{p}_k$ along the inflow and induce downwash at the locations $\mathbf{m}_k$.

4.3.2 Solving for Induced Velocities, Forces, and Moments

The Kutta-Joukowsky theorem relates circulation and lift; applied to a discrete wing segment we obtain:

$$\Gamma_k = \frac{1}{2} c_k c_{l,k}\left(\alpha_\mathrm{e}\right) V, \tag{4.7}$$

with the segment (index k) circulation Γ_k, the local chord length c_k, and the local airfoil lift coefficient $c_{l,k}$. The lift coefficient depends on the effective AoA: $\alpha_\mathrm{e} = \alpha_\mathrm{f} - \alpha_\mathrm{i}$, where α_f denotes the AoA with respect to the free stream far in front of the airplane, and α_i stands for the induced AoA. The induced downwash at position $\mathbf{m}_p$ is obtained by adding the induced speeds of all the individual vortices according to Biot-Savart

$$w_{\mathrm{i},k} = \sum_{j=1}^{n+1} \frac{\Gamma_j \mathbf{e}_V}{4\pi \|(\mathbf{p}_j - \mathbf{m}_k) \times \mathbf{e}_v\|} \left(1 + \frac{(\mathbf{p}_j - \mathbf{m}_k) \cdot \mathbf{e}_V}{\|(\mathbf{p}_j - \mathbf{m}_k)\|}\right), \tag{4.8}$$

where $\mathbf{e}_V$ stands for the (unit) direction of flight. At $\mathbf{m}_k$, the induced angle of attack is calculated as

$$\alpha_{\mathrm{i},k} = \arctan \frac{w_{\mathrm{i},k}}{V_t} \approx \frac{w_{\mathrm{i},k}}{V_t}. \tag{4.9}$$

Together with the respective 2D polar data, the above relations allow calculating the lift, drag and moment distribution (with respect to the free inflow direction $\mathbf{e}_V$) from a known circulation distribution, and can be summed and reduced to e.g. the center of mass of a whole airplane. Note that the section lift coefficient is often approximated linearly as $c_l(\alpha) \approx c_{l,0} + c_{l_\alpha}\alpha$, which allows for a direct solution when applying (4.7) and (4.8). More accurate results, in particular in the domain near maximum lift, however, are obtained by using the nonlinear section lift polar data. In this case, a standard iterative numeric solver may be used. Furthermore, airfoil data with deflected control surfaces can be included, allowing to calculate control moments (and forces).

4.4 Parametric Nonlinear Model

Let us consider the example of a simple airplane configuration with ailerons, a rudder and an elevator plus a propeller, driven by an electric motor at rotation speed ω_p.

We define the system input vector $\mathbf{u} = [\delta_a, \delta_e, \delta_r, \delta_T]$ as normalized aileron, rudder and elevator action, $\delta_a, \delta_e, \delta_r \in [-1, 1]$ as well as (normalized) thrust $\delta_T \in [0, 1]$. The fully parametric nonlinear model provided below largely follows [Ducard 2009].

4.4.1 Aerodynamics Model

We approximate the lift, drag as well as sideslip coefficients with polynomials in α and β:

$$\begin{aligned} c_L &\approx c_{L,0} + c_{L,\alpha}\alpha + c_{L,\alpha 2}\alpha^2 + c_{L,\alpha 3}\alpha^3, \\ c_D &\approx c_{D,0} + c_{D,\alpha}\alpha + c_{D,\alpha 2}\alpha^2 + c_{D,\alpha 3}\alpha^3 + c_{D,\beta 2}\beta^2, \\ c_Y &\approx c_{Y,\beta}\beta. \end{aligned} \tag{4.10}$$

Note that we deal with a slow flying airplane using an operating point relatively near to maximum lift, therefore we do not treat the c_L-α curve as linear (in contrast to [Ducard 2009] that deals with an aerobatics-type of airplane).

As far as the moments are concerned, we introduce also dependencies on normalized angular rates of the form

$${}_B\omega_n = [p_n, q_n, r_n]^T = \left[\frac{pb}{2V_t}, \frac{q\bar{c}}{2V_t}, \frac{rb}{2V_t}\right]^T. \tag{4.11}$$

A suitable approximation of the moment coefficients is now made as

$$\begin{aligned} c_l &\approx c_{l,0} + c_{l,\delta_a}\delta_a + c_{l,\beta}\beta + c_{l,p_n}p_n + c_{l,r_n}r_n, \\ c_M &\approx c_{M,0} + c_{M,\delta_e}\delta_e + c_{M,\alpha}\alpha + c_{M,q_n}q_n, \\ c_N &\approx c_{N,0} + c_{N,\delta_r}\delta_r + c_{N,\beta}\beta + c_{N,r_n}r_n. \end{aligned} \tag{4.12}$$

4.4.2 Propulsion Group Model

Finally, the propeller thrust force and friction moment needs to be modeled. With the non-dimensional coefficients c_T and c_Q we write

$$T = \rho\left(\frac{\omega_p}{2\pi}\right)^2 d^4 c_T, \tag{4.13}$$

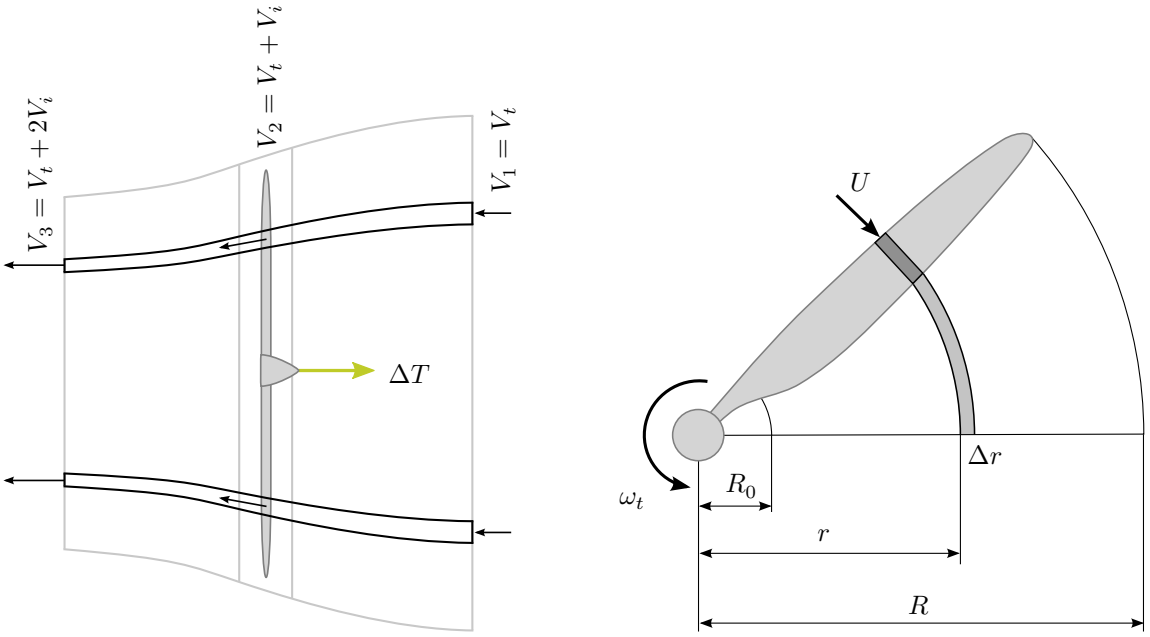
$$Q = \rho\left(\frac{\omega_p}{2\pi}\right)^2 d^5 c_Q, \tag{4.14}$$

where d denotes the propeller diameter and ω_p the rotation speed.

Blade Element Momentum Theory (BEMT)

Similar as for finite wings, accurate propeller thrust models require computation of induced velocities. A popular compromise between complexity and accuracy is provided by the Blade Element Momentum Theory (BEMT), described e.g. in [Leishman 2006] that we quickly overview in the following. Figure 4.7 introduces the geometry and

characteristic discretization of the propeller disc of radius R into annuli at radius r_k of width Δr_k. The method uses 2D aerodynamics analysis at each blade element and combines it with momentum conservation on a bounding box reaching through each of the annuli from far in front of the propeller to far behind. Typically, a root-cutout of



(a) BEMT control volume sideview.

(b) BEMT discretization of the propeller disc into annuli.

Figure 4.7: BEMT geometry cross-section side and front view: induced velocities far behind the rotor amount to twice the value at the propeller disk.

radius R_0 is considered.

Application of (incompressible) flow continuity as well as conservation of power and linear momentum through a propeller annulus yields

$$\Delta c_T^{MT} = \frac{4}{\omega_p^2 R^4} V_i (V_i + V_t) r \Delta r, \tag{4.15}$$

where V_i is the induced speed at the propeller disc. Note that the induced speed at the end of the control volume amounts to exactly double the induced speed at the propeller disc, which is a primary consequence of the aforementioned principles applied to the control volume. This furthermore explains the contraction of the stream tube annuli.

The second perspective concerns analysis of 2D flow and forces at a blade element, as depicted in Figure 4.8, using the classical decomposition into lift ΔL and drag ΔD. The thrust force ΔT and drag moment ΔQ components can thus be formulated as

$$\Delta T = N_b \left(\cos\phi \Delta L - \sin\phi \Delta D\right) \approx N_b \left(\Delta L - \phi \Delta D\right) \approx N_b \Delta L, \tag{4.16}$$

$$\Delta Q = N_b r \left(\cos\phi \Delta D + \sin\phi \Delta L\right) \approx N_b r \left(\Delta D + \phi \Delta L\right), \tag{4.17}$$

where the number of blades N_b was introduced, and where a first order (zero order, respectively) approximation for small ϕ was made. We furthermore approximate the 2D

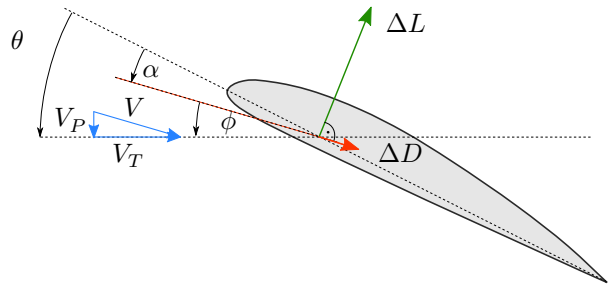


Figure 4.8: BEMT blade cross-section depicting inflow components.

lift polar as $c_l \approx c_{l,0} + c_{l,\alpha}\alpha$. Using the perpendicular inflow velocity $V_P = V_t + V_i$ as well as the tangential component $V_T = \omega_p r$, an expression for the induced velocity can now be established:

$$V_i(r, \omega_p, V_t) = \sqrt{\left(\frac{N_b c \omega_p}{16\pi} - V_t\right)^2 + \frac{N_b c r \omega_p^2 c_{l,\alpha}}{8\pi}\left(\phi + \frac{c_{l,0}}{c_{l,\alpha}}\right)} - \frac{N_b c \omega_p}{16\pi} - V_t. \tag{4.18}$$

This essentially provides the (approximate) distribution of induced velocities; now, (4.16) and (4.17) can be applied without the approximations and using the actual propeller airfoil polar. Finally, the thrust and torque increments are summed up.

Parametric Thrust and Friction Moment Models

Analogous to the aerodynamic airplane force and moment descriptions, the next step now consists of finding a parametric version of the thrust force and moment. Using the advance ratio $J = 2\pi V_t/(\omega_p d)$, we can approximate the thrust and friction moment coefficients.

$$c_T \approx c_{T,0} + c_{T,J} J + c_{T,J2} J^2, \tag{4.19}$$

$$c_Q \approx c_{Q,0} + c_{Q,J} J + c_{Q,J2} J^2. \tag{4.20}$$

Static Electric Motor Model

In order to find the relation between the control signal $\delta_T \in [0, 1]$ to the output speed, we need to investigate properties of electric motors; the Brushless DC Motors (BLDC) employed in this context are typically approximated with a DC motor model, where an input (armature) voltage $U_\mathrm{A} = \delta_T U_\mathrm{batt}$ is assumed, with the battery voltage U_batt.

In static conditions, i.e. neglecting motor dynamics, we can apply the following relation:

$$\omega_m = \frac{1}{k_\mathrm{V}}\left(U_\mathrm{A} - R_\mathrm{A}\left(\frac{Q_m}{k_\mathrm{V}} + I_0\right)\right), \tag{4.21}$$

with the no-load current I_0, the armature resistance R_A, and the motor constant k_V that are typically provided by the motor manufacturer. Note that the motor torque Q_m and motor speed ω_m might be related to the propeller with by a gearbox and an associated efficiency.

4.4.3 Complete Dynamics Equations

The modeling process starts by treating the airplane as a single rigid body affected by external forces $\mathbf{F}$ and external moments $\boldsymbol{\tau}$.

General 6D Rigid Body Dynamics

Using the Newton-Euler formalism we write the linear and angular momentum balance in body frame $\underrightarrow{\mathcal{F}}_B$ coordinates at the airplane CoG as

$$m({}_B\dot{\mathbf{v}} + {}_B\boldsymbol{\omega} \times {}_B\mathbf{v}) = {}_B\mathbf{F}, \tag{4.22}$$

$$\mathbf{I}_B\dot{\boldsymbol{\omega}} + {}_B\boldsymbol{\omega} \times (\mathbf{I}_B\boldsymbol{\omega}) = {}_B\boldsymbol{\tau}, \tag{4.23}$$

where the linear velocity vectors ${}_B\mathbf{v} = [u, v, w]^T$ and rotational velocity ${}_B\boldsymbol{\omega} = [p, q, r]^T$ are used. m represents the airplane total mass and $\mathbf{I}$ denotes its second moment of inertia.

Due to airplane symmetry w.r.t. the x-z-plane, the inertia matrix becomes

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix}. \tag{4.24}$$

The pose is obtained by integration of velocities: we represent it by the position of the airplane CoG ${}_N\mathbf{r}_B = [x, y, z]^T$ as well as by the orientation relative to an Earth fixed navigation frame $\underrightarrow{\mathcal{F}}_N$ which is considered inertial. We use the quaternion $\mathbf{q}_{NB}$ for attitude representation corresponding to the rotation matrix $\mathbf{C}_{NB}$.

In summary, we have the equations of motion

$$\begin{aligned} {}_N\dot{\mathbf{r}}_B &= \mathbf{C}_{NB}\,{}_B\mathbf{v}\,, \\ \dot{\mathbf{q}}_{NB} &= \frac{1}{2}\boldsymbol{\Omega}({}_B\boldsymbol{\omega})\mathbf{q}_{NB}\,, \\ {}_B\dot{\mathbf{v}} &= \frac{1}{m}{}_B\mathbf{F} - {}_B\boldsymbol{\omega} \times {}_B\mathbf{v}\,, \\ {}_B\dot{\boldsymbol{\omega}} &= \mathbf{I}^{-1}\left({}_B\boldsymbol{\tau} - {}_B\boldsymbol{\omega} \times \mathbf{I}_B\boldsymbol{\omega}\right), \end{aligned} \tag{4.25}$$

with the matrix $\boldsymbol{\Omega}$

$$\boldsymbol{\Omega}({}_B\boldsymbol{\omega}) = \begin{bmatrix} [{}_B\boldsymbol{\omega}]^\times & -{}_B\boldsymbol{\omega} \\ {}_B\boldsymbol{\omega}^T & 0 \end{bmatrix}, \tag{4.26}$$

where $[{}_B\boldsymbol{\omega}]^\times$ corresponds to the skew–symmetric matrix of the vector ${}_B\boldsymbol{\omega}$.

As a common alternative orientation parameterization, roll, pitch and yaw angles $\varphi \in [-\pi, \pi]$, $\theta \in [-\pi/2, \pi/2]$ and $\psi \in [-\pi, \pi]$ are also used. The rotation matrix is then composed as

$$\mathbf{C}_{NB} = \mathbf{C}_Z(\psi)\,\mathbf{C}_Y(\theta)\,\mathbf{C}_X(\varphi)\,. \tag{4.27}$$

In words, $\mathbf{C}_{NB}$ is obtained by successive rotations of ψ around the z-axis, θ around an y-axis and φ around the x-axis.

In this minimal form, the orientation dynamics can be expressed as:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix}}_{=:\mathbf{J}_r^{-1}} {}_B\boldsymbol{\omega}\,. \tag{4.28}$$

Note that the Jacobian $\mathbf{J}_r$ becomes singular at the boundaries of $\theta = \pm\pi/2$, which is why quaternions are preferred.

Summary of the Forces and Moments

For application of the 6D rigid body dynamics, the forces and moments from the various sources need to be assembled and represented in the body frame:

$$ {}_B\mathbf{F} = \begin{bmatrix} L\sin\alpha - D\cos\alpha \\ Y \\ -L\cos\alpha - D\sin\alpha \end{bmatrix} + \begin{bmatrix} T\cos\epsilon_T \\ 0 \\ T\sin\epsilon_T \end{bmatrix} + \mathbf{C}_{BN} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{4.29} $$

$$ {}_B\boldsymbol{\tau} = \begin{bmatrix} L_A \\ M_A \\ N_A \end{bmatrix} + \begin{bmatrix} L_T \\ M_T \\ N_T \end{bmatrix}, \tag{4.30} $$

where the T-subscript indicates (possible) moment components from thrust. Note that the system inputs $\mathbf{u}$ are hidden inside these forces and moments. Also be aware of α and β containing parts of the state vector:

$$ \alpha = \arctan2(w_t, u_t), \tag{4.31} $$

$$ \beta = \arcsin(v_t/V_t), \tag{4.32} $$

where the true airspeed components are used:

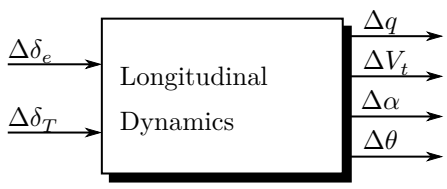
$$ {}_B\mathbf{v}_t := \begin{bmatrix} u_t \\ v_t \\ w_t \end{bmatrix} = {}_B\mathbf{v} - \mathbf{C}_{BN}\,{}_N\mathbf{w}, \tag{4.33} $$

with the wind vector ${}_N\mathbf{w}$.

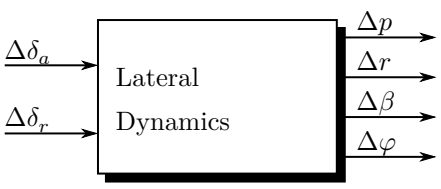
Longitudinal vs. Lateral Dynamics and Linearization

As common throughout literature, the linearized airplane dynamics are written using the roll, pitch and yaw angles, which is why we will follow the same approach. Conceptually, they could be written in a singularity-free form using a minimal quaternion deviation.

Typically, a separation into *longitudinal* and *lateral* dynamics is made, in order to assess related characteristics separately. Furthermore, the state is transformed to contain α, β, and V_t rather than ${}_B\mathbf{v}$. The linear dynamics around a reference state $\mathbf{x}_0$ and input $\mathbf{u}_0$ vector take the form $\Delta\dot{\mathbf{x}}_{\text{lon}} = \mathbf{A}_{\text{lon}}\Delta\mathbf{x}_{\text{lon}} + \mathbf{B}_{\text{lon}}\Delta\mathbf{u}_{\text{lon}}$ and $\Delta\dot{\mathbf{x}}_{\text{lat}} = \mathbf{A}_{\text{lat}}\Delta\mathbf{x}_{\text{lat}} + \mathbf{B}_{\text{lat}}\Delta\mathbf{u}_{\text{lat}}$ as visualized in Figure 4.9. The formulation employed here largely follows [Ducard 2009].



(a) Longitudinal airplane dynamics.



(b) lateral airplane dynamics.

Figure 4.9: Linearized longitudinal and lateral dynamics of $\Delta\mathbf{x}$ for inputs $\Delta\mathbf{u}$ (around $\mathbf{u}_0$ and $\mathbf{x}_0$).

The longitudinal nonlinear equations are given as:

$$\begin{aligned}
\dot{q} &= \frac{1}{I_{yy}}\left[M_A + M_T - (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2)\right], \\
\dot{V}_t &= \frac{1}{m}\left[-D\cos\beta + Y\sin\beta + T\cos(\alpha - \epsilon_T)\cos\beta + mg_1\right], \\
\dot{\alpha} &= \frac{1}{\cos\beta}\left[\frac{1}{mV_t}(-L - T\sin(\alpha - \epsilon_T) + mg_3) + q_A\right], \\
\dot{\theta} &= q\cos\varphi - r\sin\varphi,
\end{aligned} \tag{4.34}$$

and the lateral nonlinear equations amount to:

$$\begin{aligned}
\dot{p} &= \frac{I_{zz}(L_A + L_T - T_p)}{I_{xx}I_{zz} - I_{xz}^2} - \frac{I_{xz}(N_A + N_T - T_r)}{I_{xx}I_{zz} - I_{xz}^2}, \\
\dot{r} &= \frac{I_{xz}(L_A + L_T - T_p)}{I_{xx}I_{zz} - I_{xz}^2} + \frac{I_{zz}(N_A + N_T - T_r)}{I_{xx}I_{zz} - I_{xz}^2}, \\
\dot{\beta} &= -r_A + \frac{1}{mV_t}\left[Y\cos\beta + D\sin\beta - T\cos(\alpha - \epsilon_t)\sin\beta + mg_2\right], \\
\dot{\varphi} &= p + q\sin\varphi\tan\theta + r\cos\varphi\tan\theta,
\end{aligned} \tag{4.35}$$

where the following terms were used:

$$\begin{aligned}
g_1 &= g(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\varphi\cos\theta + \sin\alpha\cos\beta\cos\varphi\cos\theta), \\
g_2 &= g(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\varphi\cos\theta - \sin\alpha\sin\beta\cos\varphi\cos\theta), \\
g_3 &= g(\sin\alpha\sin\theta + \cos\alpha\cos\varphi\cos\theta), \\
q_A &= q\cos\beta - p\sin\beta\cos\alpha - r\sin\alpha\sin\beta, \\
r_A &= r\cos\alpha - p\sin\alpha, \\
T_p &= (I_{zz} - I_{yy})qr + I_{xz}pq, \\
T_r &= (I_{yy} - I_{xx})qp - I_{xz}qr.
\end{aligned} \tag{4.36}$$

The linearization of 4.34 and 4.35 is straight-forward to obtain and not provided here due to space constraints.

For a specific operating point, typically stationary with $\varphi_0 = 0$, $\theta_0 = \alpha_0$, and ${}_B\boldsymbol{\omega} = \mathbf{0}$ the standard tools of linear systems analysis can be employed. Most importantly, the *pole* locations in the imaginary plane will tell the characteristic modes and their dynamic stability.

4.5 Solar Airplane Characteristics

The modeling described so far had implications concerning the design process of senseSoar and AtlantikSolar; it furthermore forms the basis for enhanced accuracy in the state estimation as well as for model-based control and/or controller simulation and tuning. We therefore provide an examplary overview of the senseSoar model, both in terms of static as well as dynamic analysis.

4.5.1 Aerodynamics

For performance analysis, as well as the important case of longitudinal static stability, the airplane polars were generated and are shown in Figure 4.10. The nominal operating point of 10.8 m/s at an air density of 1.18 kg/m^3 was used. Also shown are the

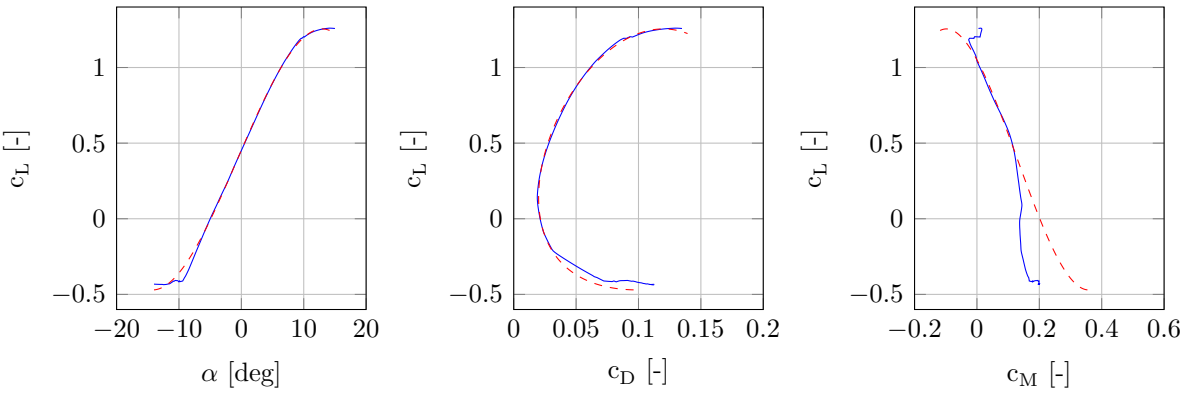


Figure 4.10: senseSoar lift, drag and pitch moment polars. The polynomial versions of the coefficients w.r.t. angle of attack ar shown for comparison as red dashed lines.

parametric versions (dashed) as described above. The chosen polynomial degrees allow for a satisfactory approximation—concerning the lift and drag values this holds for the full operation range, whereas the fit of the moment polar is slightly more limited to the neighborhood of the operating point.

Comparing the senseSoar airplane polars to the main wing airfoil polars shown in Figure 4.4, the (expected) fundamental differences become apparent: the airplane has a stable pitch moment equilibrium, achieves less lift increase per angle of attack increment, and comes with more drag—parasitic, and, additionally, induced drag. These findings confirm the usefulness of tedious modeling rather than employing a more straight-forward approach neglecting induced flows (such as described in [Noth 2008]).

In Figure 4.11, we show the variation of the crucial quantities *glide ratio* c_L/c_D and climb factor c_L^3/c_D^2 as a function of AoA α; the former has direct interpretation in motorless

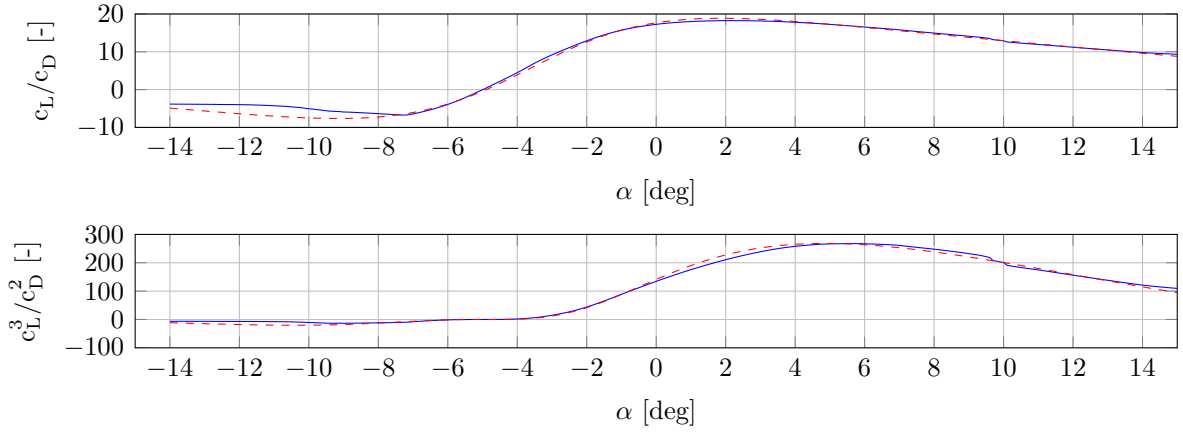


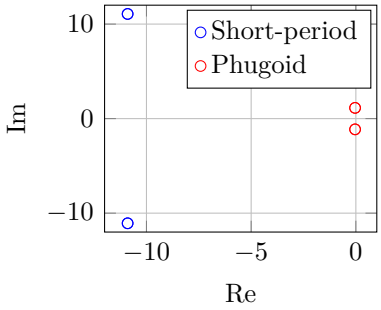
Figure 4.11: senseSoar glide ratio c_L/c_D and climb factor c_L^3/c_D^2 as a function of AoA: since solar airplanes must minimize power consumption for best performance, the ideal point corresponds to the maximum climb factor at $\alpha_0 \approx 5.5°$.

flight. The maximum glide ratio furthermore corresponds to maximum achievable range, while the maximum climb factor describes the operating point of minimum power usage, i.e. maximizing endurance (assuming constant propulsive efficiency in both cases). The latter maximum, found at $\alpha_0 \approx 5.5°$, thus describes the nominal operating point of the senseSoar airplane—which lies rather close to stall. This fact further motivates accurate modeling and control, in order to ensure both *efficient* as well as *safe* operation.

4.5.2 Aircraft Dynamics

In order to assess the main dynamics, the pole locations of the longitudinal and lateral dynamics, linearized around nominal level flight, are provided.

Figure 4.12 shows the senseSoar longitudinal poles (i.e. the eigenvalues of $\mathbf{A}_{\text{lon}}$) along with an illustration of the physical interpretation. The slower mode (Phugoid) has a



(a) Longitudinal poles of senseSoar.



(b) Phugoid mode.

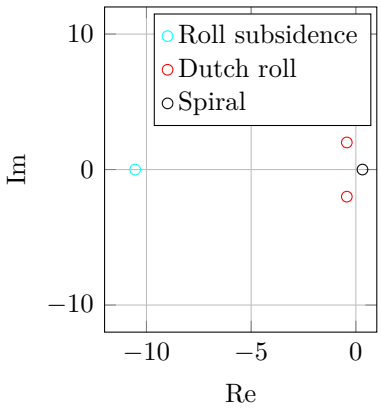


(c) Short-period mode.

Figure 4.12: Longitudinal poles of senseSoar and physical interpretation of the corresponding modes. The Phugoid frequency and damping are identified as $\omega_n = 1.14$ rad/s and $\zeta = 0.0328$; concerning the short-period mode, we have $\omega_n = 15.5$ rad/s and $\zeta = 0.701$.

frequency and damping of $\omega_n = 1.14$ rad/s and $\zeta = 0.0328$, respectively. While still stable, the damping is comparably weak. The short-period mode (AoA oscillation) is much faster with $\omega_n = 15.5$ rad/s and more damped at $\zeta = 0.701$.

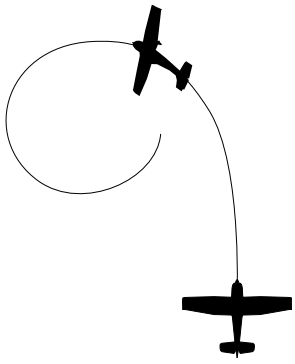
The lateral poles and corresponding modes are shown in Figure 4.13. The complex



(a) Lateral poles of senseSoar.



(b) Dutch roll mode.



(c) Spiral mode (divergent).

Figure 4.13: Lateral poles of senseSoar and physical interpretation of the related modes. The Dutch roll frequency is found at $\omega_n = 2.05$ rad/s and its damping at $\zeta = 0.211$. Roll subsidence is quite fast ($\omega_n = 10.5$ rad/s). The spiral is slightly unstable (divergent), but acceptably slow ($\omega_n = 0.316$ rad/s).

pole pair corresponds to the Dutch roll mode, a combined yaw-roll oscillation. On senseSoar, it is characterized with an eigenfrequency at $\omega_n = 2.05$ rad/s and a damping of $\zeta = 0.211$. Roll subsidence (roll response) is fast with $\omega_n = 10.5$ rad/s. The spiral mode is slightly unstable, which is not uncommon and acceptable in this case, since it is slow with $\omega_n = 0.316$ rad/s.

4.6 Discussion

A complete toolchain was presented that allows for the generation of airplane dynamic models: a 2.5D Nonlinear Lifting Line (NLL) approach uses 2D airfoil polars and the airplane geometry, in order to calculate the flow field as well as the aerodynamic forces and moments acting on the aircraft. The approach was furthermore complemented with Blade Element Momentum Theory (BEMT) combined with a static electric motor model, for thrust modeling.

A fully parametric formulation of the forces was provided, resulting in a white-box parametric nonlinear dynamic model when fed into the rigid-body dynamics (that were briefly overviewed as well). The models form an important component of the design process; related analysis of both the static as well as dynamic properties of the senseSoar solar airplane were presented as an example.

Furthermore, the models act as the basis of both accurate state estimation as shown in the next Chapter 5, as well as for model-based controller synthesis, pre-tuning, or as simulation tools in general.

Chapter 5
Robust Multi-Sensor State Estimation

State estimation forms a basic building block for mobile robot operation. In the following, a suite of algorithms for on-board estimation is proposed that fuses multiple and to some extent complementary sensor measurements. We make use of readings from a MEMS-based Inertial Measurement Unit (IMU), the Global Positioning System (GPS), static and dynamic pressure, three-axis compass (magnetometer) measurements, as well as imagery. When deployed on real platforms, typically coming with limited computational power, such estimation algorithms need to fulfill demanding specifications in terms of three major characteristics: efficiency, robustness and accuracy.

Unmanned solar airplanes impose extremely challenging constraints in this respect: starting with computational efficiency of algorithms, this is directly translatable into a power consumption share contributing to operational limitations, most notably in terms of range and flight time. Driven by the indispensable need for real-time operation, algorithms in fact have to be tailored to hardware and vice-versa. The overall power consumption will then be composed of algorithm-dependent power consumption and power expended due to hardware weight.

Apart from the need for efficiency, the complementary requirement for robustness must be given highest attention: flying systems are deployed long term in challenging environments with respect to disturbances (e.g. turbulence), and potential failure of sen-

---

Parts of this chapter appear in:

- S. Leutenegger and R. Y. Siegwart. A Low-Cost and Fail-Safe Inertial Navigation System for Airplanes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 612–618. IEEE, 2012.
- S. Leutenegger, A. Melzer, K. Alexis, and R. Y. Siegwart. Robust State Estimation for Small Unmanned Airplanes. To appear in *Systems and Control (MSC), 2014 IEEE International Multi-Conference on*. IEEE, 2014.
- S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2013.
- S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization. In *The International Journal of Robotics Research (IJRR)*, submitted

sors. We thus propose multi-sensor fusion strategies that come with inherent redundancy, allowing to some degree to detect sensor outages, and continue operation.

The third aspect, accuracy, naturally comes at some trade-off with computational demand; both autopilot and application specifications, however, set requirements for precise estimators. The proposed algorithms were thus designed to be scalable, such that higher accuracy may be gained at the expense of more computation. Last but not least, our algorithms are compared for competitiveness with respect to state-of-the-art solutions regarding the primary measure of accuracy.

This chapter outlines the approaches that were followed in order to address the challenges stated above. Regarding minimal computational complexity, a filtering based Inertial Navigation System (INS) for airplanes is presented first that does not use camera images and is still capable of handling GPS outage of arbitrary time intervals. In a second part, an algorithm based on nonlinear estimation is proposed that complements inertial sensing with camera measurements, allowing to jointly estimate the robot state, as well as its environment. In comprehensive evaluations, we assess the validity and competitiveness with respect to the state of the art.

5.1 Introduction

This section relates the proposed INS filter as well as the visual-inertial estimator to existing works in literature and states the contributions made in this context. We furthermore make an attempt to explain the parallels and differences of filtering versus nonlinear optimization.

5.1.1 Related Works

Both approaches that will be described in this Section have elements in common: first of all, they share the Inertial Measurement Unit (IMU) as a core part delivering temporal constraints. Crassidis [2005] describes IMU kinematics and modeling of internal states, such as biases, in detail. We will, as common in literature, apply a set of simplifications due to limited accuracy of the Micro-Electro-Mechanical Systems (MEMS) sensors employed, as discussed e.g. in [Shin and El-Sheimy 2004]. The second common element concerns the parameterization of orientation that is intrinsic to pose estimation in six Degrees of Freedom (DoF). A minimal orientation representation e.g. with Tait-Bryan angles yields singularities, which renders it unusable for estimation of dynamic motion patterns as present in small-scale UAS operation. The singularity-free quaternion description cannot be directly used due to the additional unit length constraint. We will use a multiplicative approach as explained in [Toda et al. 1969] and more recently in [Trawny and Roumeliotis 2005], which is widely used nowadays, e.g. also in Furgale [2011], the notation of which we follow largely throughout this work.

For linear systems, the Kalman filter [Kalman 1960] provides the maximum likelihood solution to the estimation problem, formulated in a prediction-update scheme. Pose estimation in six DoF, however, comes with the challenge of inherent nonlinearities. As an obvious generalization, the Extended Kalman Filter (EKF) uses nonlinear state propagation, while resorting to linearization in the covariance propagation and parts of the update computation. Even under the (very optimistic) assumption that both process noise and measurements can be accurately described by Gaussian distributions,

state uncertainty is not properly propagated. This may constitute a problem, if the covariances are large compared to the local linearization, e.g. when using an imprecise initialization. Consequently, several variants to the classical EKF have been developed, such as Sigma-Point Kalman Filters (SPKF), with their most prominent example, the Unscented Kalman Filter (UKF) [Julier and Uhlmann 1997].

Concerning state estimation for aerospace applications, Crassidis [2005] provides a tutorial on the complete INS kinematic equations and the application of a UKF as compared to the EKF when providing GPS position updates. Such an SPKF with position and barometric height updates was also demonstrated on an unmanned helicopter [van der Merwe and Wan 2004], demonstrating significantly better performance when the GPS is turned off for some time. An extensive comparison of the approaches mentioned is given in [Shin 2005].

In many applications, it is sufficient to only estimate the attitude or to estimate the position in a second, decoupled filtering step. For this purpose, the EKF is still widely used, also by commercial developments in the context of small unmanned airplanes (e.g. in [Jang and Liccardo 2006]). For a well-posed attitude estimation problem, two different fixed reference directions are needed. Often the Earth magnetic field vector is used as one direction and gravity as the other. The popular Complementary Filter simply low-pass filters the fixed reference directions and combines it with a high-pass filtered estimate obtained by gyro integration. Simple filters use the assumption that the long-term mean of the measured accelerations transformed into an inertial reference frame will correspond to the gravity vector. This assumption holds for some motion types, but certainly not for airplane flight, due to possibly long lasting centripetal accelerations e.g. when circling. Recently, this shortcoming was mitigated by using the airspeed measurement along with rotation measurements in order to estimate centripetal accelerations within the Complementary Filter framework [Euston et al. 2008]. A detailed survey of attitude estimation methods is provided in [Crassidis et al. 2007].

In the context of unmanned airplane state estimation, a plethora of approaches have been proposed; Johnson et al. [2001] use only GPS measurements combined with the coordinated-turn assumption to form a minimalistic technique for orientation estimation. Formulating aerodynamics with rigid-body dynamics offers a further interesting source of information that can be used in the estimation process: in [Lievens et al. 2005], fusion of this model with GPS is suggested and proven to yield observable orientation, while tracking wind remains a challenge. An approach of also integrating IMU measurements along with a dynamic airplane model and GPS is suggested in [Kumon et al. 2005], which allows for estimating wind in a loosely coupled process that estimates attitude separately. A more expensive approach is suggested in [Langelaan et al. 2011] that models a wind field subjected to temporal as well as spatial constraints; such a method is rather suitable for offline processing than for online-estimation. The use of dynamic pressure measurements from a pitot tube for wind estimation is an obvious choice which is made e.g. in [Roberts et al. 2005] that uses GPS velocities as updates to a lateral wind model with slow dynamics in a separate process from attitude dynamics. In earlier work [Leutenegger and Siegwart 2012], we used airspeed measurements for wind estimation without including it into the state and performed backup airspeed updates to the EKF, whenever GPS was unavailable.

Integrating also camera measurements with an IMU offers two intriguingly strong benefits: first of all, the visual constraints deliver spatial constraints, complementing the temporal ones as obtained from the IMU. Secondly, it allows for the perception of the

environment, which may constitute an absolute necessity for many robotic applications, including UAS operation close to the terrain. The vision-only algorithms which form the foundation for today's Visual-Inertial Navigation Systems (VINS) can be categorized into opimization-based Structure-from-Motion (SfM) and filtering based methods. Due to computational constraints, for a long time, Vision-based real-time odometry or Simultaneous Localization and Mapping (SLAM) algorithms such as those presented in Davison [2003] were only possible using a filtering approach. Subsequent research [Strasdat et al. 2010], however, has shown that nonlinear optimization based approaches, as commonly used for offline SfM, can provide better accuracy for a similar computational cost when compared to filtering approaches, given that the structural sparsity of the problem is preserved. Henceforth, it has been popular to maintain a rather sparse graph of keyframes and associated landmarks subject to nonlinear optimizations [Klein and Murray 2007].

The earliest results in VINS originate from the work of Jung and Taylor [2001] for (spline based) batch and of Chai et al. [2002]; Roumeliotis et al. [2002] for filtering based approaches. Subsequently, a variety of filtering based approaches have been published based on EKFs [Kim and Sukkarieh 2007; Mourikis and Roumeliotis 2007; Li and Mourikis 2012b; Weiss et al. 2012; Lynen et al. 2013], Iterated EKFs (IEKFs) [Strelow and Singh 2004, 2003] and Unscented Kalman Filters (UKFs) [Shin and El-Sheimy 2004; Ebcin and Veth 2007; Kelly and Sukhatme 2011] to name a few, which over the years showed an impressive improvement in precision and a reduction of computational complexity. Today such 6 DoF visual-inertial estimation systems can be run online on consumer mobile devices [Li and Mourikis 2012c; Li et al. 2013].

In order to limit computational complexity, many works follow the *loosely coupled* approach. Konolige et al. [2011a] integrate IMU measurements as independent inclinometer and relative yaw measurements into an optimization problem using stereo vision measurements. In contrast, Weiss et al. [2012] use vision-only pose estimates as updates to an EKF with indirect IMU propagation. Similar approaches can be followed for loosely coupled batch based algorithms such as in Ranganathan et al. [2007] and Indelman et al. [2012], where relative stereo pose estimates are integrated into a factor-graph with non-linear optimization including inertial terms and absolute GPS measurements. It is well known that loosely coupled approaches are inherently sub-optimal since they disregard correlations amongst internal states of different sensors, as opposed to *tightly coupled* methods.

A notable contribution in the area of filtering based VINS is the work of Mourikis and Roumeliotis [2007] who proposed an EKF based real-time tightly coupled fusion using monocular vision, called the Multi-State Constraint Kalman Filter (MSCKF) which performs nonlinear-triangulation of landmarks from a set of camera poses over time before using them in the EKF update. This contrasts with other works that only use visual constraints between pairwise camera poses [Bayard and Brugarolas 2005]. Mourikis and Roumeliotis [2007] also show how the correlations between errors of the landmarks and the camera locations—which are introduced by using the estimated camera poses for triangulation—can be eliminated and thus result in an estimator which is consistent and optimal up to linearization errors. Another monocular visual-inertial filter was proposed by Jones and Soatto [2011], presenting results on a long outdoor trajectory including IMU to camera calibration and loop closure. Li and Mourikis [2013] showed that further increases in the performance of the MSCKF are attainable by switching between the landmark processing model, as used in the MSCKF, and the full estimation of landmarks, as employed by EKF-SLAM.

Further improvements and extensions to both loosely and tightly-coupled filtering based approaches include an alternative rotation parameterization [Li and Mourikis 2012a], inclusion of rolling shutter cameras [Jia and Evans 2012; Li et al. 2013], offline [Lobo and Dias 2007; Mirzaei and Roumeliotis 2007, 2008] and online [Weiss et al. 2012; Kelly and Sukhatme 2011; Jones and Soatto 2011; Dong-Si and Mourikis 2012] calibration of the relative position and orientation of camera and IMU.

In order to benefit from increased accuracy offered by re-linearization in batch optimization, recent work focused on approximating the batch problem in order to allow real-time operation. Approaches to keep the problem tractable for online-estimation can be separated into three groups [Nerurkar et al. 2013]: Firstly, incremental approaches, such as the factor-graph based algorithms by Kaess et al. [2012]; Bryson et al. [2009], apply incremental updates to the problem while factorizing the associated information matrix of the optimization problem or the measurement Jacobian into square root form [Bryson et al. 2009; Indelman et al. 2012]. Secondly, fixed-lag smoother or sliding-window filter approaches [Dong-Si and Mourikis 2011; Sibley et al. 2010; Huang et al. 2011] consider only poses from a fixed time interval in the optimization. Poses and landmarks which fall outside the window are marginalized with their corresponding measurements being dropped. Forming non-linear constraints between different optimization parameters in the marginalization step unfortunately destroys the sparsity of the problem, such that the window size has to be kept fairly small for real-time performance. The smaller the window, however, the smaller the benefit of repeated re-linearization. Thirdly, keyframe based approaches preserve sparsity by maintaining only a subset of camera poses and landmarks while they discard (rather than marginalize) intermediate quantities. Nerurkar et al. [2013] present an efficient offline Maximum *A Posteriori* (MAP) algorithm which uses all information from non-keyframes and landmarks to form constraints between keyframes by marginalizing a set of frames and landmarks without impacting the sparsity of the problem. While this form of marginalization shows small errors when compared to the full batch MAP estimator, we target a version with a fixed window size suitable for online and real-time operations. In this thesis and our previous work [Leutenegger et al. 2013], we therefore drop measurements from non-keyframes and marginalize the respective state. When keyframes drop out of the window over time, we marginalize the respective states and some landmarks commonly observed to form a (linear) prior for a remaining sub-part of the optimization problem. Our approximation scheme strictly keeps the sparsity of the original problem. This is in contrast to e.g. Sibley et al. [2010], who accept some loss of sparsity due to marginalization. The latter sliding window filter, in a visual-inertial variant, is used for comparison in Li and Mourikis [2012b]: it proves to perform better than the original MSCKF, but interestingly, an improved MSCKF variant using first-estimate Jacobians yields even better results. We aim at performing similar comparisons between an MSCKF implementation—that includes the use first estimate Jacobians—and our keyframe as well as optimization based algorithm.

Apart from the differentiation between batch and filtering approaches, it has been a major interest to increase the estimation accuracy by studying the observability properties of VINS. There is substantial work on the observability properties given a particular combination of sensors or measurements [Martinelli 2011; Weiss 2012] or only using data from a reduced set of IMU axes [Martinelli 2014]. Global unobservability of yaw and position, as well as growing uncertainty with respect to an initial pose of reference are intrinsic to the visual-inertial estimation problem [Hesch et al. 2012a; Huang et al. 2013; Hesch et al. 2013]. This property is therefore of particular interest when comparing filtering

approaches to batch-algorithms: the representation of pose and its uncertainty in a global frame of reference usually becomes both numerically and conceptually problematic as the uncertainty for parts of the state undergoes unbounded growth by nonlinear dynamics, while remaining low for the observable sub parts of the state. Our batch approach therefore uses a formulation of relative uncertainty of keyframes to avoid expressing global uncertainty.

Unobservability of the VINS problem poses a particular challenge to filtering approaches where repeated linearization is typically not possible: Huang et al. [2009] have shown that these linearization errors may erroneously render parts of the estimated state numerically observable. Hesch et al. [2012b] and others [Huang et al. 2011; Kottas et al. 2012; Hesch et al. 2012a, 2013; Huang et al. 2013] derived formulations allowing to choose the linearization points of the VINS system in a way such that the observability properties of the linearized and non-linear system are equal. In our proposed algorithm, we employ first-estimate Jacobians, i.e. whenever linearization of a variable is employed, we fix the linearization point for any subsequent linearization involving that particular variable.

5.1.2 Outline and Contributions of the Estimators

In relation to the literature described above, we propose two algorithms for state estimation that make the following contributions.

First, we propose a multi-sensor fusion framework based on the EKF for airplane state estimation that uses indirect IMU propagation and updates from GPS, magnetometer, as well as static and dynamic pressure measurements.

- The filter tracks position, orientation, speed and IMU biases, as well as wind and pressure at sea level in a tightly-coupled manner. A model of the airplane aerodynamics is used for enhanced accuracy.
- We implemented the filter on-board glider avionics and on our UAS autopilot micro-controller boards, underlining the real-time capability with constrained computational resources.
- We obtain robustness with respect to GPS outages over an extended amount of time, still allowing for unbiased orientation and airspeed vector estimation. To our knowledge, the proposed approach is the first to employ the aforementioned sensors to achieve these characteristics during GPS outages of arbitrary length and in the presence of sustained accelerations (circling).
- Extensive results with a manned glider demonstrate the applicability and robustness of the proposed method.

A second estimator addresses the inclusion of visual cues into the estimation process. The method is based on nonlinear estimation rather than filtering applied to a window of keyframes that co-observe landmarks and that are temporally linked by inertial terms.

- We employ the keyframe paradigm by marginalization for drift-free estimation also when slow or no motion at all is present: rather than using an optimization window of time-successive poses, the kept keyframes may be spaced arbitrarily far in time, keeping visual constraints—while still respecting an IMU term. Our

formulation of relative uncertainty of keyframes allows for building a pose graph without expressing global pose uncertainty, taking inspiration from RSLAM [Mei et al. 2011].

- We provide a fully probabilistic derivation of IMU error terms, including the respective information matrix, relating successive image frames without explicitly introducing states at IMU-rate.

- We present a generic N-camera ($N \geq 1$, i.e. including the mono case) Visual-Inertial Odometry (VIO) framework. We overview initialization and bootstrapping; in the stereo-version, the performance will gradually transform into the monocular case when the ratio between camera baseline and distance to structure becomes small. Tight fusion of IMU and keypoint measurements in nonlinear optimization allows for enhanced outlier filtering and application of m-estimators.

- After having shown the superior performance of the suggested method compared to a loosely-coupled approach in Leutenegger et al. [2013], we present extensive evaluation results with respect to a stochastic cloning sliding window filter (following the MSCKF implementation of Mourikis et al. [2009] which includes the use of first-estimate Jacobians) in terms of accuracy on different motion profiles. Our algorithm consistently outperforms the filtering-based method, while it admittedly incurs higher computational cost. To the best of our knowledge, such a direct comparison of visual-inertial state estimation algorithms, as suggested by different research groups, is novel to the field.

- We present the formulation for online camera extrinsics estimation that may be applied after standard intrinsics calibration. Evaluation results demonstrate the applicability of this method, when initializing with inaccurate camera pose estimates with respect to the IMU.

- We make an attempt to present our work to a level of detail that would allow the reader to re-implement our framework.

- Various datasets featuring individual characteristics in terms of motion, appearance, and scene depth were recorded ranging from hand-held indoor motion to bicycle riding.

The remainder of this chapter is structured as follows: this Section is ended in 5.1.3 with a formal analysis of filtering versus nonlinear optimization. Section 5.2 introduces the notations and definitions used throughout this work. The element common to all estimators presented here, namely the IMU model, is subsequently detailed in Section 5.3. We introduce the light-weight INS filtering framework for airplanes that does not rely on camera measurements in Section 5.4. In Section 5.5, we turn to employing also keypoint measurements from cameras combined with inertial measurements in a nonlinear optimization. Section 5.6 finally discusses the main elements and findings of the suggested sensor fusion approaches and indicates ways of extending and applying them.

5.1.3 Filtering Versus Nonlinear Optimization

Since filtering and optimization are competing approaches to sensor fusion that are both employed in this work, we want to provide a short overview on the respective formulations and their differences.

As a starting point for both cases, we consider a state $\mathbf{x}_{\mathrm{R}}(t)$, and some form of (control) input $\mathbf{u}(t)$ that are subjected to continuous time nonlinear stochastic dynamics

$$\dot{\mathbf{x}}_{\mathrm{R}}(t) = \mathbf{f}_c(\mathbf{x}_{\mathrm{R}}(t), \mathbf{u}(t), \mathbf{w}_c(t)), \tag{5.1}$$

where $\mathbf{w}_c(t)$ denotes a multi-dimensional Gaussian white noise process.

We furthermore introduce a minimal error state $\delta\boldsymbol{\chi}(t)$ around a linearization point $\bar{\mathbf{x}}(t)$. Now, the linearized error dynamics take the form

$$\delta\dot{\boldsymbol{\chi}}(t) \approx \mathbf{F}_c(\bar{\mathbf{x}}_{\mathrm{R}}(t), \bar{\mathbf{u}}(t))\delta\boldsymbol{\chi}(t) + \mathbf{G}_c(\bar{\mathbf{x}}_{\mathrm{R}}(t), \bar{\mathbf{u}}(t))\mathbf{w}_c(t) + \mathbf{B}_c(\bar{\mathbf{x}}_{\mathrm{R}}(t), \bar{\mathbf{u}}(t))\mathbf{u}(t), \tag{5.2}$$

where the Jacobian matrices evaluated at $\bar{\mathbf{x}}(t)$ as well as at a nominal input $\bar{\mathbf{u}}(t)$ were introduced:

$$\mathbf{F}_c := \frac{\partial \mathbf{f}_c}{\partial \delta\boldsymbol{\chi}}, \ \mathbf{G}_c := \frac{\partial \mathbf{f}_c}{\partial \mathbf{w}_c}, \text{ and } \mathbf{B}_c := \frac{\partial \mathbf{f}_c}{\partial \mathbf{u}}. \tag{5.3}$$

Using a discrete-time version of the above, we obtain

$$\begin{aligned} \mathbf{x}_{\mathrm{R}}^{p+1} &= \mathbf{f}_d(\mathbf{x}_{\mathrm{R}}^p, \mathbf{u}^p, \mathbf{w}^p), \\ \delta\boldsymbol{\chi}_{\mathrm{R}}^{p+1} &\approx \mathbf{F}_d(\bar{\mathbf{x}}_{\mathrm{R}}^p, \bar{\mathbf{u}}^p)\delta\boldsymbol{\chi}_{\mathrm{R}}^p + \mathbf{G}_d(\bar{\mathbf{x}}_{\mathrm{R}}^p, \bar{\mathbf{u}}^p)\mathbf{w}^p + \mathbf{B}_d(\bar{\mathbf{x}}_{\mathrm{R}}^p, \bar{\mathbf{u}}^p)\mathbf{u}^p, \end{aligned} \tag{5.4}$$

where $\mathbf{w}^p$ is normally distributed around zero with covariance $\mathbf{Q}^p$.

When using inertial measurements, concretely angular rates and acceleration (including acceleration due to gravity), we have two options of using them. In the *indirect* formulation, they can be regarded as inputs $\mathbf{u}$ of the dynamics, leading to noise characteristics to be assigned to $\mathbf{w}$ that are directly related with IMU noise parameters. Using the *direct* formulation, they act as true measurements—this however requires to formulate higher order dynamics $\mathbf{f}_c$, that essentially act as a motion prior which may or may not be related to physical properties of the system to be estimated. In order to limit the estimator complexity, and to avoid biasing, we prefer and use the indirect formulation throughout this work.

In filtering, we propagate the mean of the state estimate, $\bar{\mathbf{x}}_{\mathrm{R}}$, and some form of uncertainty of $\delta\boldsymbol{\chi}_{\mathrm{R}}$ (i.e. the covariance matrix in a classical EKF) using Equation (5.4). At the same time, in classical filtering, the old states and respective parts of the uncertainty are marginalized out, since they are not supposed to be used anymore. To complement the propagated, *a priori* estimate, residuals are formulated of the form

$$\mathbf{y}_i = \mathbf{z}_i - \mathbf{h}(\mathbf{x}_{\mathrm{R}}), \tag{5.5}$$

where $\mathbf{z}_i$ is normally distributed with covariance $\mathbf{R}_i$. The linear approximation of $\mathbf{y}_i$ around $\bar{\mathbf{x}}_{\mathrm{R}}$ that plays a major role in many filtering approaches is obtained as:

$$\mathbf{y}_i(\mathbf{z}_i, \mathbf{x}_{\mathrm{R}}) \approx \mathbf{z}_i - \mathbf{H}(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_{\mathrm{R}} - \mathbf{h}(\bar{\mathbf{x}}_{\mathrm{R}}), \tag{5.6}$$

where we introduced the Jacobian matrix

$$\mathbf{H} := \frac{\partial \mathbf{h}}{\partial \delta\boldsymbol{\chi}_{\mathrm{R}}}. \tag{5.7}$$

In an update process, the least-squares *a posteriori* estimates are subsequently computed, along with the posterior uncertainty. In some cases, due to non-invertible nonlinearities, such a residual cannot be formulated; rather we write it as an error term $\mathbf{e}_i(\mathbf{x}_\mathrm{R}, \mathbf{z}_i)$, the linearization of which becomes

$$\mathbf{e}_i(\mathbf{z}_i, \mathbf{x}_\mathrm{R}) \approx \mathbf{e}_i(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_\mathrm{R}) + \mathbf{E}_z(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_\mathrm{R})\delta\mathbf{z} + \mathbf{E}_x(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_\mathrm{R})\delta\boldsymbol{\chi}_\mathrm{R}, \tag{5.8}$$

where $\mathbf{z}$ is also written with a linearization point $\bar{\mathbf{z}}$ and a disturbance $\delta\mathbf{z}$. $\mathbf{E}_x$ denotes the Jacobian matrix w.r.t. $\delta\boldsymbol{\chi}_\mathrm{R}$, and $\mathbf{E}_z$ the one w.r.t. $\delta\mathbf{z}$.

Nonlinear optimization, however, does not know the notion of propagation; it simply seeks to *iteratively* minimize a cost function J that is formed as a sum of weighted least squares errors $\mathbf{e}_i$:

$$\begin{aligned} J(\mathbf{x}) &= \sum_i \mathbf{e}_i(\mathbf{x}, \mathbf{z}_i)^T \mathbf{W}_i \mathbf{e}_i(\mathbf{x}, \mathbf{z}_i), \\ \mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}}\, J(\mathbf{x}), \end{aligned} \tag{5.9}$$

with the error term likelihood

$$f(\mathbf{e}_i | \mathbf{x} = \bar{\mathbf{x}}) = \mathcal{N}(\mathbf{0}, \mathbf{W}_i^{-1}), \tag{5.10}$$

where $\mathbf{W}_i$ can thus be calculated from the measurement covariance $\mathbf{R}_i$ as

$$\mathbf{W}_i = \left(\mathbf{E}_z \mathbf{R}_i \mathbf{E}_z^T\right)^{-1}. \tag{5.11}$$

This formulation allows the introduction of temporal error terms (see 5.5) by inserting time-consecutive states $\mathbf{x}_\mathrm{R}^k$ and $\mathbf{x}_\mathrm{R}^{p+P}$ as part of the set of variables $\mathbf{x}$ compared to predictions using $\mathbf{f}_d(\mathbf{x}^p, \mathbf{u}^p, \mathbf{w}^p)$. Note that in contrast to filtering, this grows the amount of variables linearly over time, since no past states are marginalized out; if marginalization is needed, it can be applied, as will be detailed in 5.5, but requires prior linearization of nonlinear error terms linked to the states to be marginalized out.

5.2 Prerequisites

Throughout this work, an effort is made to employ the following consistent notations, definitions, and conventions.

5.2.1 Notation and Definitions

$\underrightarrow{\mathcal{F}}_A$ denotes a reference frame A; a point P represented in frame $\underrightarrow{\mathcal{F}}_A$ is written as position vector ${}_A\mathbf{r}_P$, or ${}_A\boldsymbol{r}_P$ when in homogeneous coordinates. A transformation between frames is represented by a homogeneous transformation matrix $\boldsymbol{T}_{AB}$ that transforms the coordinate representation of homogeneous points from $\underrightarrow{\mathcal{F}}_B$ to $\underrightarrow{\mathcal{F}}_A$. Its rotation matrix part is written as $\mathbf{C}_{AB}$; the corresponding quaternion is written as $\mathbf{q}_{AB} = [\boldsymbol{\epsilon}^T, \eta]^T \in S^3$, $\boldsymbol{\epsilon}$ and η representing the imaginary and real parts. We adopt the notation introduced in [Barfoot et al. 2011]: concerning the quaternion multiplication $\mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC}$, we introduce a left-hand side compound operator $[.]^+$ and a right-hand side operator $[.]^\oplus$ that output matrices such that $\mathbf{q}_{AC} = [\mathbf{q}_{AB}]^+ \mathbf{q}_{BC} = [\mathbf{q}_{BC}]^\oplus \mathbf{q}_{AB}$. Taking velocity as an example of a physical quantity represented in frame $\underrightarrow{\mathcal{F}}_A$ that relates frame $\underrightarrow{\mathcal{F}}_B$ and $\underrightarrow{\mathcal{F}}_C$, we write ${}_A\mathbf{v}_{BC}$, i.e. the velocity of frame $\underrightarrow{\mathcal{F}}_C$ as seen from $\underrightarrow{\mathcal{F}}_B$.

5.2.2 Frames

The pose estimation problem consists of finding the transformation between a fixed reference frame and a *body frame* $\underrightarrow{\mathcal{F}}_B$. Various sensors may be attached to the body, with respective coordinate frames, most importantly the *IMU* $\underrightarrow{\mathcal{F}}_S$, and *cameras* $\underrightarrow{\mathcal{F}}_{C_i}$.

Whenever consistent positioning in a global reference is sensible, we use the latitude, longitude and altitude $\mathbf{p} = [\phi, \lambda, h]^T$ referring to the *World Geodetic System 1984 (WGS 84) Earth ellipsoid coordinates*. We furthermore use a locally tangential *North, East, Down (NED)* frame $\underrightarrow{\mathcal{F}}_N$ for representation of vector entities as well as to provide a reference for describing orientation.

In the purely visual or visual-inertial problem, global referencing needs to be replaced by expressing position and orientation with respect to a *World-fixed* frame $\underrightarrow{\mathcal{F}}_W$ that may be placed and oriented arbitrarily, except for the z-axis that needs to be aligned with gravitational acceleration.

5.2.3 States and Local Parameterization

Depending on the type of estimator, we compose time varying states as well as static variables to be estimated. Let $\mathbf{x}$ be the set of variables that is currently being estimated. This work focuses on the estimation of various poses (i.e. the robot, but also sensors), 3D positions of landmarks, as well as velocities and sensor-intrinsic states such as IMU biases.

More formally, we represent the robot position as ${}_W\mathbf{r}_S$ in the World frame, or as $\mathbf{p} = [\phi, \lambda, h]^T$. Orientation is described either w.r.t. $\underrightarrow{\mathcal{F}}_W$, if global yaw is unobservable, or $\underrightarrow{\mathcal{F}}_N$ otherwise, i.e. $\mathbf{q}_{WS}$ or $\mathbf{q}_{NS}$, respectively. The j^{th} landmark is represented in homogeneous (World) coordinates: ${}_W\boldsymbol{r}_{L_j} =: {}_W\boldsymbol{l}^j \in \mathbb{R}^4$. At this point, however, we set its fourth component to one.

In general, the states live in a manifold, therefore we use a perturbation in tangent space $\mathfrak{g}$ and employ the group operator $\boxplus$, that is not commutative in general, the exponential exp(.) and logarithm log(.). Now, we can define the perturbation $\delta\mathbf{x} := \mathbf{x} \boxplus \bar{\mathbf{x}}^{-1}$ around the estimate $\bar{\mathbf{x}}$. We use a minimal coordinate representation $\delta\boldsymbol{\chi} \in \mathbb{R}^{\dim \mathfrak{g}}$. A bijective mapping $\Phi : \mathbb{R}^{\dim \mathfrak{g}} \to \mathfrak{g}$ transforms from minimal coordinates to tangent space. Thus we obtain for as transformations from and to minimal coordinates:

$$\delta\mathbf{x} = \exp(\Phi(\delta\boldsymbol{\chi})), \tag{5.12}$$

$$\delta\boldsymbol{\chi} = \Phi^{-1}(\log(\delta\mathbf{x})). \tag{5.13}$$

Variables living in Euclidean space (such as velocities and biases) employ a simple perturbation that is applied via + operation. The non-Euclidean cases are outlined in the following.

We use the minimal (3D) axis-angle perturbation of orientation $\delta\boldsymbol{\alpha} \in \mathbb{R}^3$ which can be converted into its quaternion equivalent $\delta\mathbf{q}$ via the exponential map:

$$\delta\mathbf{q} := \exp\left(\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 0 \end{bmatrix}\right) = \begin{bmatrix} \operatorname{sinc}\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\|\frac{\delta\boldsymbol{\alpha}}{2} \\ \cos\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\| \end{bmatrix}. \tag{5.14}$$

Therefore, using the group operator $\otimes$, we write $\mathbf{q}_{WS} = \delta\mathbf{q} \otimes \bar{\mathbf{q}}_{WS}$. Note that linearization of the exponential map around $\delta\boldsymbol{\alpha} = \mathbf{0}$ yields:

$$\delta\mathbf{q} \approx \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\alpha} \\ 1 \end{bmatrix} = \boldsymbol{\iota} + \frac{1}{2}\begin{bmatrix} \mathbf{I}_3 \\ \mathbf{0}_{1\times 3} \end{bmatrix}\delta\boldsymbol{\alpha}, \tag{5.15}$$

with the unit quaternion $\boldsymbol{\iota}$.

As landmark perturbation, we use a simple Euclidean version $\delta\boldsymbol{\beta} \in \mathbb{R}^3$ that is applied as $\delta\boldsymbol{l}^j := [\delta\boldsymbol{\beta}^T, 0]^T$ by addition.

5.3 IMU Kinematics and Bias Dynamics

As a common building block for all the estimators to be described below, we employ a rather standard model of IMU kinematics along with modeled dynamics of IMU biases. The assumption is made that the Earth turn rate can be neglected—which is reasonable when using MEMS IMUs, the accuracy of which is comparably low. In the following, the non-linear continuous time model is provided first, followed by the linearized error state dynamics and the discrete time approximation.

5.3.1 Non-Linear Continuous Time Model

Formally, we describe the dynamics of robot position, orientation, velocity, as well as gyro bias $\mathbf{b}_\mathrm{g}$ and accelerometer bias $\mathbf{b}_\mathrm{a}$. Using a World frame reference for position ${}_W\mathbf{r}_S$ and orientation $\mathbf{q}_{WS}$ representation, as well as expressing velocities in a robo-centric fashion ${}_S\mathbf{v}_{WS}$ (in short ${}_S\mathbf{v}$), we obtain:

$$\begin{aligned}
{}_W\dot{\mathbf{r}}_S &= \mathbf{C}_{WS}\,{}_S\mathbf{v}\,, \\
\dot{\mathbf{q}}_{WS} &= \frac{1}{2}\boldsymbol{\Omega}\left({}_S\boldsymbol{\omega}\right)\mathbf{q}_{WS}\,, \\
{}_S\dot{\mathbf{v}} &= {}_S\tilde{\mathbf{a}} + \mathbf{w}_{c,\mathrm{a}} - \mathbf{b}_\mathrm{a} + \mathbf{C}_{SW}\,{}_W\mathbf{g} - \left({}_S\boldsymbol{\omega}\right)\times {}_S\mathbf{v}\,, \\
\dot{\mathbf{b}}_\mathrm{g} &= \mathbf{w}_{c,b_\mathrm{g}}, \\
\dot{\mathbf{b}}_\mathrm{a} &= -\frac{1}{\tau}\mathbf{b}_\mathrm{a} + \mathbf{w}_{c,b_\mathrm{a}}.
\end{aligned} \tag{5.16}$$

where the elements of $\mathbf{w}_c := [\mathbf{w}_{c,\mathrm{g}}^T, \mathbf{w}_{c,\mathrm{a}}^T, \mathbf{w}_{c,b_\mathrm{g}}^T, \mathbf{w}_{c,b_\mathrm{a}}^T]^T$ are each uncorrelated zero-mean Gaussian white noise processes. ${}_S\tilde{\mathbf{a}}$ are accelerometer measurements and ${}_W\mathbf{g}$ represents the Earth's gravitational acceleration vector. In contrast to the gyro bias modeled as random walk, we use the time constant $\tau > 0$ to model the accelerometer bias as bounded random walk. The matrix $\boldsymbol{\Omega}$ is formed from the estimated angular rate ${}_S\boldsymbol{\omega} = {}_S\tilde{\boldsymbol{\omega}} + \mathbf{w}_{c,\mathrm{g}} - \mathbf{b}_\mathrm{g}$, with gyro measurement ${}_S\tilde{\boldsymbol{\omega}}$:

$$\boldsymbol{\Omega}\left({}_S\boldsymbol{\omega}\right) := \begin{bmatrix} -{}_S\boldsymbol{\omega} \\ 0 \end{bmatrix}^\oplus. \tag{5.17}$$

Alternatively, we may prefer to choose world-centric velocities ${}_W\mathbf{v}_{WS}$, which change the IMU kinematics part of (5.16) to:

$$\begin{aligned}
{}_W\dot{\mathbf{r}}_S &= {}_W\mathbf{v}\,, \\
\dot{\mathbf{q}}_{WS} &= \frac{1}{2}\boldsymbol{\Omega}\left({}_S\boldsymbol{\omega}\right)\mathbf{q}_{WS}\,, \\
{}_W\dot{\mathbf{v}} &= \mathbf{C}_{WS}\left({}_S\tilde{\mathbf{a}} + \mathbf{w}_{c,\mathrm{a}} - \mathbf{b}_\mathrm{a}\right) + {}_W\mathbf{g}.
\end{aligned} \tag{5.18}$$

When referenced in WGS 84 coordinates, we can express the position, orientation and velocity dynamics as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{J}_{NN}\mathbf{v}\,, \\ \dot{\mathbf{q}}_{WS} &= \frac{1}{2}\mathbf{\Omega}\left({}_S\boldsymbol{\omega}\right)\mathbf{q}_{NS}\,, \\ {}_N\dot{\mathbf{v}} &= \mathbf{C}_{NS}\left({}_S\tilde{\mathbf{a}} + \mathbf{w}_{c,\mathrm{a}} - \mathbf{b}_{\mathrm{a}}\right) + {}_N\mathbf{g}, \end{aligned} \tag{5.19}$$

where world-centric velocities were used, but the robot-centric version (5.16) could be adopted without change. The Jacobian $\mathbf{J}_N$ transforms velocities in $\underrightarrow{\mathcal{F}}_N$ to derivatives of the ellipsoid coordinates:

$$\mathbf{J}_N = \begin{bmatrix} \frac{1}{R_\phi + h} & 0 & 0 \\ 0 & \frac{1}{(R_\lambda + h)\cos(\phi)} & 0 \\ 0 & 0 & -1 \end{bmatrix}, \tag{5.20}$$

where the local Earth radii R_ϕ and R_λ are obtained with the WGS 84 semimajor axis a and eccentricity e:

$$\begin{aligned} R_\phi &= \frac{a(1-e^2)}{(1-e^2\sin^2\phi)^{\frac{3}{2}}}, \\ R_\lambda &= \frac{a}{(1-e^2\sin^2\phi)^{\frac{1}{2}}}. \end{aligned} \tag{5.21}$$

5.3.2 Linearized Model of the Error States

The linearized version of the above Equation (5.16) around the current estimate $\overline{\mathbf{x}}_R$ will play a major role both the filtering formulation as well as nonlinear estimation with marginalization. We therefore review it here briefly: the error dynamics take the form

$$\delta\dot{\boldsymbol{\chi}}_R \approx \mathbf{F}_c(\overline{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_R + \mathbf{G}_c(\overline{\mathbf{x}}_{\mathrm{R}})\mathbf{w}_c, \tag{5.22}$$

where $\mathbf{G}_c$ is straightforward to derive and:

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \left[\overline{\mathbf{C}}_{WS}\,{}_S\overline{\mathbf{v}}\right]^\times & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\overline{\mathbf{C}}_{SW}\left[{}_W\mathbf{g}\right]^\times & -\left[{}_S\overline{\boldsymbol{\omega}}\right]^\times & -\left[{}_S\overline{\mathbf{v}}\right]^\times & -\mathbf{1}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{\tau}\mathbf{1}_3 \end{bmatrix}, \tag{5.23}$$

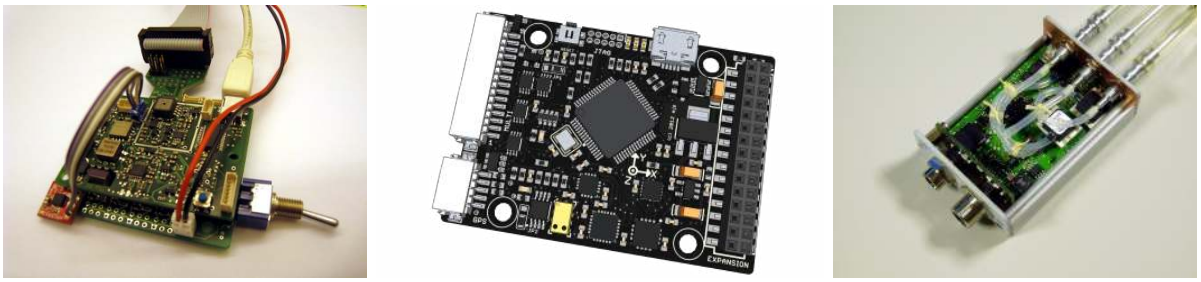
$[.]^\times$ denoting the skew-symmetric cross-product matrix associated with a vector. Overbars stand for evaluation of the respective symbols with current estimates.

The linearized dynamics of the error states in the world-centric representation is found to be:

$$\mathbf{F}_{c,W} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \overline{\mathbf{C}}_{WS} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \left[\overline{\mathbf{C}}_{WS}\left({}_S\tilde{\mathbf{a}} - \overline{\mathbf{b}}_{\mathrm{a}}\right)\right]^\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\overline{\mathbf{C}}_{WS} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{\tau}\mathbf{1}_3 \end{bmatrix}. \tag{5.24}$$

5.4 Multi-Sensor Fusion without Cameras

A reliable and precise state estimation is crucial for any kind of unmanned airplanes. The Extended Kalman Filter (EKF) widely deployed in aerospace state estimation has been the workhorse fusing inertial sensor feeds with navigation aids for decades. With the advent of increasingly precise, high bandwidth and low noise MEMS inertial sensors and three-axis electronic compasses during the last few years, such state estimation filters may be run on-board low-cost electronics. Thus also the autonomous robotics community showed a rising interest in such systems. In Figure 5.1 we show the board developed at the Autonomous Systems Lab (ASL), the px4 autopilot[2], as well as the Integrated Sensor Unit (ISU) by the avionics developer Triadis—on all of which we implemented our algorithm.



(a) ASL board. (b) px4 autopilot. (c) Triadis ISU.

Figure 5.1: The platforms the proposed filter was implemented on: the left an central one are in use on-board the ASL unmanned airplanes, and the right one is part of commercially available (manned) glider avionics.

In the following, we present a full EKF INS filter that is robust with respect to GPS outage as well as dynamic maneuvers, including long-term circling. We use updates obtained by magnetometers, static pressure and position in normal operation, and airspeed updates during GPS loss. Therefore, the covariances are staying small compared to nonlinearities, thus both operation modes form scenarios where the EKF has a long history of success and where its performance is equivalent to newer alternatives such as the UKF.

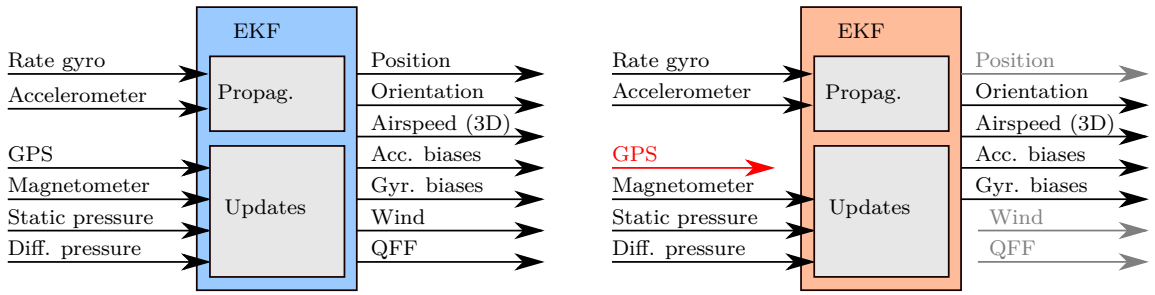
We explain the underlying INS kinematic model along with the two proposed filtering modes in Section 5.4.1. State reconstruction results obtained with GPS as well as during outage are provided in Section 5.4.2: the data is recorded on a manned glider during several hours of flight.

5.4.1 Estimation Filter Operation

The working principle of the EKF framework to be detailed in the following is outlined in Figure 5.2.

The indirect EKF uses angular rates and acceleration measurements from the IMU for propagation. Measurements are formulated with GPS (if available), static and differential pressure, as well as with a 3D magnetic compass. During GPS outage as caused e.g. by a steep turn, airspeed, orientation and IMU biases are still being tracked by the filter,

[2]See pixhawk.org/modules/px4fmu, accessed March 2014.



(a) Normal operation.

(b) GPS outage condition.

Figure 5.2: Operation of the suggested filter in normal mode (left) and during GPS outage (right): it uses accelerations and angular rate measurements for propagation, and GPS (if available), pressure measurements and a magnetometer for updating. During normal operation, the 3D wind is estimated as part of the states; in an outage condition, airspeed, orientation, as well as IMU biases are still tracked.

while lateral wind estimation as well as pressure at sea level (QFF) is disabled for obvious reasons; the output of the filter will, however, still allow an autopilot to operate, even if the position estimates are subjected to growing uncertainties.

In the following, we provide the mathematical formulation of the EKF propagation as well as updates with the various measurements arriving at different rates. We finally also describe how GPS outages and outliers in any of the measurements are handled.

Notation, Frames and States

Concretely, we consider the sensor frame $\underrightarrow{\mathcal{F}}_S$ in which angular rates, accelerations and the magnetic field is measured by the IMU. As a reference, we use a local North-East-Down (NED) navigation frame $\underrightarrow{\mathcal{F}}_N$ the origin of which is always coincident with the one of $\underrightarrow{\mathcal{F}}_S$. We express the global position in terms of latitude, longitude and altitude $\mathbf{p} = [\phi, \lambda, h]^T$ referring to the WGS 84 Earth ellipsoid coordinates. We furthermore introduce an airplane frame $\underrightarrow{\mathcal{F}}_B$ located at its center of gravity.

We set up the estimation problem of tracking the state vector

$$\mathbf{x}_\mathrm{R} = \left[\mathbf{p}^T, \mathbf{q}_{NS}^T, {}_N\mathbf{v}_{SN}^T, \mathbf{b}_\mathrm{g}^T, \mathbf{b}_\mathrm{a}^T, {}_N\mathbf{d}^T, p_0\right]^T \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^{13}, \tag{5.25}$$

with the orientation quaternion $\mathbf{q}_{NS}$, the velocity ${}_N\mathbf{v}_{SN}$ (written in short as ${}_N\mathbf{v}$). $\mathbf{b}_\mathrm{g}$ and $\mathbf{b}_\mathrm{a}$ denote the biases of rate gyros and accelerometers, respectively. ${}_N\mathbf{d}$ stands for the three-dimensional wind vector, and p_0 for the pressure at mean sea level (QFF). Overall, we write the error state vector as

$$\delta\boldsymbol{\chi}_\mathrm{R} = [\delta\mathbf{p}^T, \delta\boldsymbol{\alpha}^T, \delta\mathbf{v}^T, \delta\mathbf{b}_\mathrm{g}^T, \delta\mathbf{b}_\mathrm{a}^T, \delta\mathbf{d}^T, \delta p_0]^T \in \mathbb{R}^{19}. \tag{5.26}$$

System Dynamics and IMU Kinematics

For propagation of the state $\mathbf{x}_\mathrm{R}$ and the error state covariance matrix $\mathbf{P}_\mathrm{R}$, the standard EKF formulation employed needs the system dynamics to be expressed. We use the indirect IMU propagation model detailed in Section 5.3, the state vector of which was

augmented with the wind vector ${}_N\mathbf{d}$ and QFF (pressure at sea level) p_0:

$$\begin{aligned}
\dot{\mathbf{p}} &= \mathbf{J}_{NN}\mathbf{v}\,, \\
\dot{\mathbf{q}}_{WS} &= \frac{1}{2}\mathbf{\Omega}\left({}_S\boldsymbol{\omega}\right)\mathbf{q}_{NS}\,, \\
{}_N\dot{\mathbf{v}} &= \mathbf{C}_{NS}\left({}_S\tilde{\mathbf{a}} + \mathbf{w}_{c,\mathrm{a}} - \mathbf{b}_{\mathrm{a}}\right) + {}_N\mathbf{g} \\
\dot{\mathbf{b}}_{\mathrm{g}} &= \mathbf{w}_{c,b_{\mathrm{g}}}, \\
\dot{\mathbf{b}}_{\mathrm{a}} &= -\frac{1}{\tau}\mathbf{b}_{\mathrm{a}} + \mathbf{w}_{c,b_{\mathrm{a}}}, \\
{}_N\dot{\mathbf{d}} &= \mathbf{w}_{c,\mathrm{d}} \\
\dot{p}_0 &= w_{c,\mathrm{p}}.
\end{aligned} \tag{5.27}$$

The matrix $\mathbf{\Omega}$ is formed from the estimated angular rate ${}_S\boldsymbol{\omega} = {}_S\tilde{\boldsymbol{\omega}} + \mathbf{w}_{c,\mathrm{g}} - \mathbf{b}_{\mathrm{g}}$, with gyro measurement ${}_S\tilde{\boldsymbol{\omega}}$:

$$\mathbf{\Omega}\left({}_S\boldsymbol{\omega}\right) := \begin{bmatrix} -{}_S\boldsymbol{\omega} \\ 0 \end{bmatrix}^{\oplus}. \tag{5.28}$$

We use the Gaussian white noise vector $\mathbf{w_c} := [\mathbf{w}_{c,\mathrm{g}}^T, \mathbf{w}_{c,\mathrm{a}}^T, \mathbf{w}_{c,b_{\mathrm{g}}}^T, \mathbf{w}_{c,b_{\mathrm{a}}}^T, \mathbf{w}_{c,\mathrm{d}}^T, w_{c,\mathrm{p}}]^T$.

The linearized error dynamics around $\bar{\mathbf{x}}_{\mathrm{R}}$ take the form

$$\delta\dot{\boldsymbol{\chi}}_R \approx \mathbf{F}_c(\bar{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_{\mathrm{R}} + \mathbf{G}(\bar{\mathbf{x}}_{\mathrm{R}})\mathbf{w}_c, \tag{5.29}$$

where $\mathbf{G}$ is straightforward to derive and:

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \overline{\mathbf{C}}_{NS} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \left[\overline{\mathbf{C}}_{WS}\left({}_S\tilde{\mathbf{a}} - \bar{\mathbf{b}}_{\mathrm{a}}\right)\right]^{\times} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\overline{\mathbf{C}}_{NS} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\frac{1}{\tau}\mathbf{1}_3 & \mathbf{0}_{3\times4} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times4} \end{bmatrix}. \tag{5.30}$$

The above equations are taken into their discrete-time representations by simple Euler forward integration using the time interval Δt, in order to save computation. But conceptually, any integration scheme may be applied.

GPS Updates

Using the above state representations and assuming the GPS antenna is close to the IMU, updates with both position measurements $\tilde{\mathbf{p}}$ and velocity measurements (which are currently not used) ${}_N\tilde{\mathbf{v}}$ become trivial (linear), i.e. the residuals are formulated as $\mathbf{y}_{\mathrm{p}} = \tilde{\mathbf{p}} - \mathbf{H}_{\mathrm{p}}\mathbf{x}_{\mathrm{R}}$ for the position and $\mathbf{y}_{\mathrm{v}} = {}_N\tilde{\mathbf{v}} - \mathbf{H}_{\mathrm{v}}\mathbf{x}_{\mathrm{R}}$. The matrices $\mathbf{H}_{\mathrm{p}}$ and $\mathbf{H}_{\mathrm{v}}$ consist of zeros and one identity three-by-three block respectively.

Magnetometer Updates

We measure the magnetic flux density vector in $\underrightarrow{\mathcal{F}}_S$, i.e. we can use the measurement ${}_S\tilde{\mathbf{b}}$. The theoretical magnetic vector ${}_N\mathbf{b}$ is calculated using the IGRF-11 model[3] as a

[3] See www.ngdc.noaa.gov/IAGA/vmod/igrf.html, accessed March 2014.

function of $\mathbf{p}$ and time. A naive residual term could be simply formulated using the difference or cross-product of the two (normalized) vectors as we used it in previous work [Leutenegger and Siegwart 2012]. In order to minimize biasing (mostly affecting roll and pitch) that may be caused by inaccuracies of the IGRF-11 model, we apply a series of transformations to formulate a residual:

$$e_{\mathrm{m}} = {}_N\mathbf{e}_{\mathrm{z}}^T \left(\mathbf{n}(\mathbf{\Pi}_{\mathrm{h}} \mathbf{C}_{NS}\, {}_S\tilde{\mathbf{b}}) \times \mathbf{n}(\mathbf{\Pi}_{\mathrm{h}} {}_N\mathbf{b}) \right), \tag{5.31}$$

where the projections ${}_N\mathbf{e}_{\mathrm{z}}^T := [0, 0, 1]$ and $\mathbf{\Pi}_{\mathrm{h}} := \mathrm{diag}([1, 1, 0])$ were used, and where $\mathbf{n}(\mathbf{a})$ denotes the normalization operation of an arbitrary vector $\mathbf{a}$ to unit length. In words, this is the sine of the (signed) angle between the projections into the $\underrightarrow{\mathcal{F}}_N$ horizontal plane of measured and theoretical magnetic vectors. The Jacobians are found as

$$\begin{aligned} \frac{\partial e_{\mathrm{m}}}{\partial \delta \boldsymbol{\chi}_{\mathrm{R}}} &= \begin{bmatrix} \mathbf{0}_{1\times 3} & {}_W\mathbf{e}_{\mathrm{z}}^T \left[-\mathbf{n}(\mathbf{\Pi}_{\mathrm{h}} {}_N\mathbf{b}) \right]^{\times} \mathbf{N}(\mathbf{\Pi}_{\mathrm{h}} \tilde{\mathbf{b}}) \mathbf{\Pi}_{\mathrm{h}} \left[\overline{\mathbf{C}}_{NS}\, {}_S\tilde{\mathbf{b}} \right]^{\times} & \mathbf{0}_{1\times 13} \end{bmatrix}, \\ \frac{\partial e_{\mathrm{m}}}{\partial {}_S\tilde{\mathbf{b}}} &= {}_W\mathbf{e}_{\mathrm{z}}^T \left[-\mathbf{n}(\mathbf{\Pi}_{\mathrm{h}} {}_N\mathbf{b}) \right]^{\times} \mathbf{N}(\mathbf{\Pi}_{\mathrm{h}} \tilde{\mathbf{b}}) \mathbf{\Pi}_{\mathrm{h}} \overline{\mathbf{C}}_{NS}, \end{aligned} \tag{5.32}$$

with the Jacobian of the normalization

$$\mathbf{N}(\mathbf{a}) := \frac{\partial \mathbf{n}(\mathbf{a})}{\partial \mathbf{a}} = \frac{[\mathbf{a}]^{\times} [-\mathbf{a}]^{\times}}{\|\mathbf{a}\|^3}. \tag{5.33}$$

Note that the residual e_{m} is not formulated in the standard EKF measurement form. Its covariance will in fact have to be calculated from the error state covariance $\mathbf{P}_{\mathrm{R}}$ and the measurement covariance $\mathbf{R}_{\mathrm{b}}$ as:

$$s_{\mathrm{b}} = \frac{\partial e_{\mathrm{m}}}{\partial \delta \boldsymbol{\chi}_{\mathrm{R}}} \mathbf{P}_{\mathrm{R}} \frac{\partial e_{\mathrm{m}}}{\partial \delta \boldsymbol{\chi}_{\mathrm{R}}}^T + \frac{\partial e_{\mathrm{m}}}{\partial {}_S\tilde{\mathbf{b}}} \mathbf{R}_{\mathrm{b}} \frac{\partial e_{\mathrm{m}}}{\partial {}_S\tilde{\mathbf{b}}}^T. \tag{5.34}$$

The equivalent linearized observation model needed to compute updates simply uses $\mathbf{H}_{\mathrm{m}} = -\frac{\partial e_{\mathrm{m}}}{\partial \delta \boldsymbol{\chi}_{\mathrm{R}}}$.

Pressure Updates with Aerodynamics Constraints

Our formulation of measurements involving static pressure p_{s} and dynamic pressure p_{d} in fact also requires outside air temperature measurements $\tilde{T}_{\mathrm{h}}$ as well as accelerometer readings ${}_S\tilde{\mathbf{a}}$. We thus use a combined measurement vector $\mathbf{z}_{\mathrm{a}} = [\tilde{p}_{\mathrm{s}}, \tilde{p}_{\mathrm{d}}, \tilde{T}_{\mathrm{h}}, {}_S\tilde{\mathbf{a}}^T]^T$. A static pressure measurement can be formulated in a straightforward manner. In order to obtain most accurate results using the dynamic pressure, however, our approach makes use of knowledge on aerodynamic properties of the airplane. Figure 5.3 overviews some geometric definitions as well as the main forces and moments acting on an airplane. More details are provided in Chapter 4. The body frame $\underrightarrow{\mathcal{F}}_B$ is introduced according to the aerospace standard: front-right-down. Note the introduction of the Angle of Attack (AoA) α with respect to the airspeed as well as lift L, drag D and side-slip force Y attacking at the airplane center of gravity. We write the airspeed vector transformed into the airplane frame $\underrightarrow{\mathcal{F}}_B$ as

$${}_B\mathbf{v}_t := \mathbf{C}_{BS}\, \mathbf{C}_{SN}\, ({}_N\mathbf{v} - {}_N\mathbf{d}) =: \begin{bmatrix} u_t \\ v_t \\ w_t \end{bmatrix}, \tag{5.35}$$

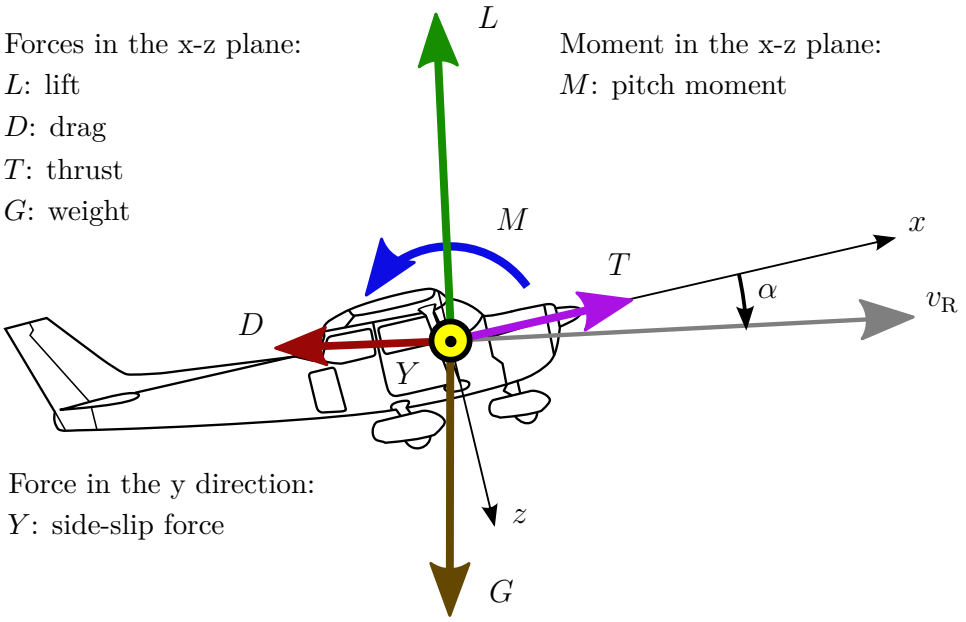


Figure 5.3: Basic geometry definitions, forces and moments acting on an airplane.

and keep in mind that its Jacobians w.r.t. the involved states amount to:

$$\begin{aligned}
\frac{\partial_B \mathbf{v}_t}{\partial \delta \boldsymbol{\alpha}} &= -\mathbf{C}_{BS}\, \overline{\mathbf{C}}_{SN}\, \left[{}_N\overline{\mathbf{v}} - {}_N\overline{\mathbf{d}} \right]^{\times}, \\
\frac{\partial_B \mathbf{v}_t}{\partial_N \mathbf{v}} &= \mathbf{C}_{BS}\, \overline{\mathbf{C}}_{SN}\,, \\
\frac{\partial_B \mathbf{v}_t}{\partial_N \mathbf{d}} &= -\mathbf{C}_{BS}\, \overline{\mathbf{C}}_{SN}\,.
\end{aligned} \tag{5.36}$$

Static pressure measurement. The static pressure model for the altitude above mean sea level h_{M} (at a constant offset to h) is set up as:

$$p_{\mathrm{h}} = \mathrm{p}_0 \left(\frac{\tilde{T}_{\mathrm{h}} - L_0 h_{\mathrm{M}}}{\tilde{T}_{\mathrm{h}}} \right)^k, k = \frac{g_0 M}{R L_0}, \tag{5.37}$$

using the temperature lapse rate $L_0 = -0.0064\mathrm{K/m}$, the gravitational acceleration at sea level g_0, M denoting the molar mass of air, and R the universal gas constant. As they will be later used, let us define the following Jacobians:

$$\begin{aligned}
\frac{\partial p_{\mathrm{h}}}{\partial h} &= -p_0 k \frac{L_0}{\tilde{T}_{\mathrm{h}}} \left(\frac{\tilde{T}_{\mathrm{h}} - L_0 h_{\mathrm{M}}}{\tilde{T}_{\mathrm{h}}} \right)^{k-1}, \\
\frac{\partial p_{\mathrm{h}}}{\partial p_0} &= \left(\frac{\tilde{T}_{\mathrm{h}} - L_0 h_{\mathrm{M}}}{\tilde{T}_{\mathrm{h}}} \right)^{k}, \\
\frac{\partial p_{\mathrm{h}}}{\partial \tilde{T}_{\mathrm{h}}} &= p_0 k \left(\frac{1}{\tilde{T}_{\mathrm{h}}} + \frac{\tilde{T}_{\mathrm{h}} - L_0 h_{\mathrm{M}}}{\tilde{T}_{\mathrm{h}}^2} \right) \left(\frac{\tilde{T}_{\mathrm{h}} - L_0 h_{\mathrm{M}}}{\tilde{T}_{\mathrm{h}}} \right)^{k-1}.
\end{aligned} \tag{5.38}$$

Differential pressure measurement. The differential pressure model follows basic fluid dynamic properties:

$$p_d = \frac{1}{2} \rho \| {}_B\mathbf{v}_t \|^2, \tag{5.39}$$

where the air density ρ is calculated as

$$\rho = \frac{\tilde{p}_\mathrm{h}}{R\tilde{T}_\mathrm{h}}. \tag{5.40}$$

Again, a Jacobian will be important:

$$\frac{\partial p_\mathrm{d}}{\partial_B \mathbf{v}_t} = \rho_B \overline{\mathbf{v}}_t^T. \tag{5.41}$$

Note that due to minor influence of temperature and static pressure measurements on this residual (compared to actual differential pressure measurement inaccuracy), we will neglect the respective Jacobians.

Aerodynamic force balances. In general, the estimation of aerodynamic force and moment coefficient is a highly complex undertaking and depending on many static and dynamic parameters, such as Reynolds number, the AoA, the airplane configuration, rotation rates, control surface position,... As a simplification in the context of state estimation, we make use of the lift as well as drag polars, i.e. $c_\mathrm{L}(\alpha)$ and $c_\mathrm{L}(c_\mathrm{D})$ that are typically available as a product of simulation and/or wind tunnel experiments. We furthermore consider the influence of the side-slip coefficient c_Y as a function of side-slip angle β.

For simplicity in formulating error terms and implementing them, we do not use the aforementioned polars directly, but transform them, in order to obtain $c_K(\alpha')$ and $c_Y(\beta')$ with $\alpha' = w/u$ and $\beta' = v/u$, where c_K is the coefficient of the aerodynamic force into the direction perpendicular to the thrust vector (i.e. typically approximately the negative airplane z-direction).

The force balances in K and Y direction are written as:

$$\begin{aligned} e_K &= m_B \mathbf{e}_K^T \mathbf{C}_{BS} \left({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a}\right) - \frac{1}{2}\rho A c_K(\alpha') \|\mathbf{\Pi}_{\mathrm{l}B}\mathbf{v}_t\|^2, \\ e_Y &= m_B \mathbf{e}_Y^T \mathbf{C}_{BS} \left({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a}\right) - \frac{1}{2}\rho A c_Y(\beta') \|\mathbf{\Pi}_{\mathrm{l}B}\mathbf{v}_t\|^2, \end{aligned} \tag{5.42}$$

where m denotes the airplane mass, A the wing surface, and $\mathbf{\Pi}_\mathrm{l} = \mathrm{diag}([1,0,1])$ projects the velocity vector into the airplane symmetry plane. ${}_B\mathbf{e}_K^T$ and ${}_B\mathbf{e}_Y^T$ stand for projections into K and Y directions.

Again, the Jacobians of these residuals w.r.t. temperature and static pressure are neglected. Using $\gamma \in \{\alpha', \beta'\}$ and $i \in \{K, Y\}$, the following Jacobians, however, are needed:

$$\begin{aligned} \frac{\partial e_i}{\partial_B \mathbf{v}_t} &= -\rho A \left(\frac{1}{2} \|\mathbf{\Pi}_{\mathrm{l}B}\overline{\mathbf{v}}_t\|^2 \frac{dc_i(\gamma)}{d\gamma} \frac{\partial \gamma}{\partial_B \mathbf{v}_t} + c_i(\gamma)\, {}_B\overline{\mathbf{v}}_t^T \mathbf{\Pi}_\mathrm{l} \right), \\ \frac{\partial e_i}{\partial \mathbf{b}_\mathrm{a}} &= -m_B \mathbf{e}_i^T \mathbf{C}_{BS}, \\ \frac{\partial e_i}{\partial_S \tilde{\mathbf{a}}} &= m_B \mathbf{e}_i^T \mathbf{C}_{BS}. \end{aligned} \tag{5.43}$$

In the practical implementation, we actually use a polynomial of degree 3 for c_K, and a simple proportionality relation for c_Y (i.e. without zero-order coefficient due to symmetry of the airplane).

For robust operation, we furthermore use a replacement for the above aerodynamics residual (5.42) whenever stall is detected, i.e. the AoA exceeds maximum or minimum angles. This error term simply implements a pseudo-measurement of the active maximum or minimum AoA and is not further detailed here. Note, however, that along with the application of this replacement residual a stall warning is issued. In automatic operation, this situation must be strictly avoided.

Also note that the accelerometer readings are used both for the aerodynamics measurement as well as in the propagation; so far, the correlation is neglected for simplicity, but in principle, it could be included by a slight adaptation of the standard EKF formulation.

Alternative to lift balance for manned gliders. Since parts of our evaluation is performed on manned gliders, for which the $c_L(\alpha)$ polynomial cannot be accessed, we propose an alternative formulation to the lift balance: we employ the so-called velocity polars $v_\mathrm{v}(v_\mathrm{f})$ that provide downward velocity v_v as a function of forward velocity v_f, typically in the form of a quadratic function. This is the most crucial description of glider performance, thus available for virtually any existing glider. As the velocity polar refers to static conditions and is referenced to the gravity direction, we compare v_v to the component of ${}_B\mathbf{v}_t$ along the measured gravity direction:

$$e_w = v_\mathrm{v}\left(\sqrt{\frac{2\tilde{p}_d}{\rho}}\right) + {}_B\mathbf{v}_t^T \mathbf{\Pi}_\mathrm{l} \mathbf{C}_{BS}\, \mathbf{n}({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a}). \tag{5.44}$$

Again, we neglect the Jacobians w.r.t. T_h and p_0. The remaining Jacobians amount to

$$\begin{aligned}
\frac{\partial e_w}{\partial {}_B\mathbf{v}_t} &= \mathbf{\Pi}_\mathrm{l}\mathbf{n}^T({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a})\mathbf{C}_{SB}\,, \\
\frac{\partial e_w}{\partial \mathbf{b}_\mathrm{a}} &= -{}_B\mathbf{v}_t^T \mathbf{\Pi}_\mathrm{l} \mathbf{C}_{BS}\, \mathbf{N}({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a}), \\
\frac{\partial e_w}{\partial {}_S\tilde{\mathbf{a}}} &= {}_B\mathbf{v}_t^T \mathbf{\Pi}_\mathrm{l} \mathbf{C}_{BS}\, \mathbf{N}({}_S\tilde{\mathbf{a}} - \mathbf{b}_\mathrm{a}), \\
\frac{\partial e_w}{\partial \tilde{p}_d} &= v_\mathrm{v}'\left(\sqrt{\frac{2\tilde{p}_d}{\rho}}\right)\frac{2}{\sqrt{2\rho\tilde{p}_d}}.
\end{aligned} \tag{5.45}$$

Assembling the residual and Jacobians. To summarize, we assemble a pressure and aerodynamics residual term as

$$\mathbf{e}_\mathrm{a} = \left[\tilde{p}_\mathrm{s} - p_\mathrm{s}, \tilde{p}_\mathrm{d} - p_\mathrm{d}, e_K, e_Y\right]^T. \tag{5.46}$$

In the case of estimation with a manned glider, we simply replace e_K with e_w. The Jacaobian w.r.t. the state becomes

$$\frac{\partial \mathbf{e}_\mathrm{a}}{\partial \delta \boldsymbol{\chi}_\mathrm{R}} = \mathbf{J}_1\mathbf{J}_2 = -\mathbf{H}_\mathrm{a}, \tag{5.47}$$

with

$$\mathbf{J}_1 = \begin{bmatrix} \begin{bmatrix} 0 & 0 & -\frac{\partial p_\mathrm{h}}{\partial h} \end{bmatrix} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & -\frac{\partial p_\mathrm{h}}{\partial p_0} \\ \mathbf{0}_{1\times 3} & -\frac{\partial p_\mathrm{d}}{\partial {}_B\mathbf{v}_t} & \mathbf{0}_{1\times 3} & 0 \\ \mathbf{0}_{1\times 3} & \frac{\partial e_K}{\partial {}_B\mathbf{v}_t} & \frac{\partial e_K}{\partial \mathbf{b}_\mathrm{a}} & 0 \\ \mathbf{0}_{1\times 3} & \frac{\partial e_Y}{\partial {}_B\mathbf{v}_t} & \frac{\partial e_Y}{\partial \mathbf{b}_\mathrm{a}} & 0 \end{bmatrix},$$

$$\mathbf{J}_2 = \begin{bmatrix} \mathbf{1}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 3} & \frac{\partial {}_B\mathbf{v}_t}{\partial \delta\boldsymbol{\alpha}} & \frac{\partial {}_B\mathbf{v}_t}{\partial {}_N\mathbf{v}} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \frac{\partial {}_B\mathbf{v}_t}{\partial {}_N\mathbf{d}} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{1}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{5.48}$$

The Jacobian w.r.t. the measurement vector $\mathbf{z}_\mathrm{a}$—needed again for residual covariance computation—is straightforward to derive.

Handling of GPS Outages and Outliers

Thanks to redundancy in the system as described above, GPS outages can be handled in a natural way with only minor adaptations. Since a sub-part of the states will become unobservable, we constrain the wind ${}_N\mathbf{d}$ to a bounded random walk in this case. Furthermore, we replace the current wind estimate with an averaged one (over some minutes before the outage occurred), in order to minimize position drift during outage.

Moreover, we set up an outlier detection scheme based on Mahalanobis gates. Be $\mathbf{S}_i$ the residual covariance of residual $\mathbf{e}_i$, we apply the consistency check of $\chi_i^2 = \mathbf{e}_i^T \mathbf{S}_i^{-1} \mathbf{e}_i$. Currently, the bounds are set to $\chi_i^2 < 25$ for measurements to be accepted for processing. Note that this check is different to checking simply the 2-norm of residuals in the sense that it respects state uncertainty, which is expected to grow, when many measurements are rejected. It thus inherently allows for recovery from overly aggressive outlier filtering. Outliers in the acceleration and rate gyro measurements are currently only handled with a simple derivative check.

5.4.2 Evaluation on Manned Glider Avionics

We have implemented the outlined algorithm both on an autopilot board to be used for the lab's unmanned solar airplane as well as on a next-generation commercial glider avionics hardware by Triadis AG. In the following, we present the state reconstruction on real glider flight data lasting several hours and involving challenging motion—such as circling for several minutes. A pilot's view snapshot (see Figure 5.4) shows the on-board implementation producing an artificial horizon output.



Figure 5.4: Glider pilot's view: on-board filter implementation, attitude and heading rendered in real-time as artificial horizon.

5.4.3 Overview of the Hardware

The main parameters, including identified process noise densities as well as measurement standard deviations are summarized in Table 5.1. Note the difference in the horizontal and

Characteristic	Symbol	Value	Unit
Gyro noise density	$\sigma_{c,\mathrm{g}}$	1.7e-4	$\mathrm{rad}/(\mathrm{s}\sqrt{\mathrm{Hz}})$
Gyro drift noise density	$\sigma_{c,\mathrm{b_g}}$	1.0e-4	$\mathrm{rad}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Acceleremoter noise density	$\sigma_{c,\mathrm{a}}$	0.008	$\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Acceleremoter drift noise density	$\sigma_{c,\mathrm{b_a}}$	1.6e-4	$\mathrm{m}/(\mathrm{s}^3\sqrt{\mathrm{Hz}})$
Acceleremoter time constant	τ	18000	s
Wind drift noise density	$\sigma_{c,\mathrm{b_a}}$	0.3	$\mathrm{m}/(\mathrm{s}\sqrt{\mathrm{Hz}})$
GPS position noise NED	$\boldsymbol{\sigma}_{\mathrm{p}}$	$[2.2, 2.2, 10]$	m
Magnetometer noise	σ_{m}	1	μT
Static pressure noise	σ_{s}	10.0	Pa
Dynamic pressure noise	σ_{d}	10.0	Pa

Table 5.1: Triadis ISU characteristics.

vertical overall GPS position accuracy. Since the vertical dynamics are better constrained than the lateral ones (due to the static pressure measurements), we can afford assuming the vertical GPS uncertainty to be significantly larger.

Note that a calibration of misalignment, scale and temperature is not yet performed. The results shown below are assumed to become considerably more precise after the application of calibration.

Remarks on Implementation

The state propagation runs at 100 Hz. Magnetometer and pressure updates as well as the wind estimation are calculated at 21Hz. GPS updates are used whenever available, typically at 4Hz. Our on-board implementation uses Matlab's C-code generator that proved to be efficient enough and allows for an extremely fast transition from simulation to running the filter on the actual hardware.

With the measured airspeed $\tilde{v}_t$, initialization of orientation is performed using the two fixed directions ${}_B\tilde{\mathbf{b}}$ as well as gravity guessed as ${}_B\tilde{\mathbf{g}} = {}_B\tilde{\mathbf{a}} - {}_B\tilde{\boldsymbol{\omega}} \times [\tilde{v}_t, 0, 0]^T$—which will produce even reasonable guesses when switched on in-flight during a dynamic maneuver (or simply circling). This property is highly important for robust operation.

Filter Results

Figure 5.5 shows a 45-minute-long trajectory segment of a flight lasting several hours. It contains contiguous circling at bank angles in the order of 45° during up to 6 minutes, generating centripetal accelerations of the same magnitude as acceleration due to gravity. For the evaluation of the proposed filtering scheme, the GPS positions were ignored for 30 minutes. We plot all GPS positions as well as the reconstructed paths. Note that the position error growth is at most linear in time in the worst case of a constant offset between the assumed wind and the actual one. The GPS was switched on after 30 minutes of simulated denial, corresponding to a distance traveled of 54.2 km without position information: the accumulated position error amounted to 750 m in this case.

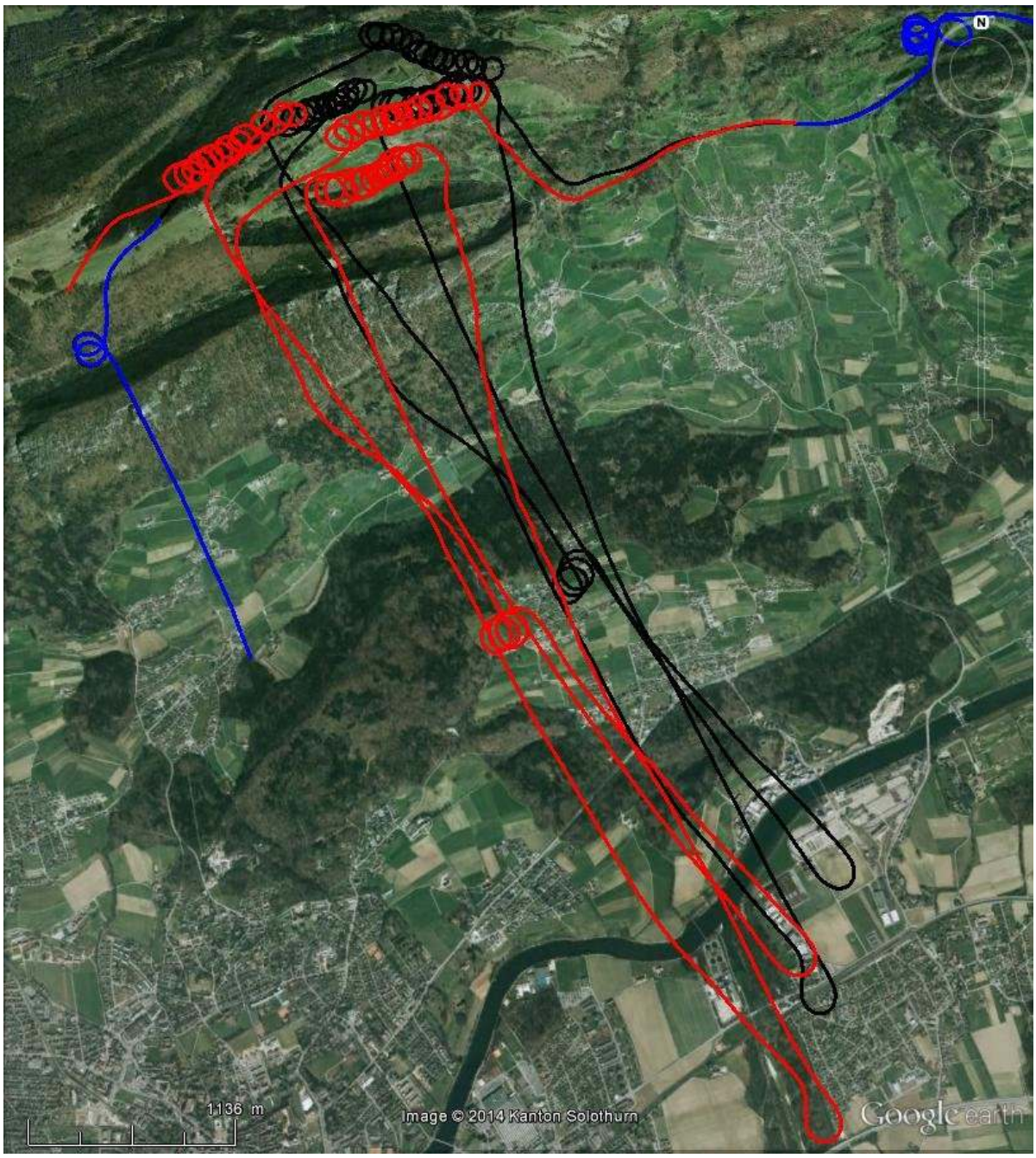


Figure 5.5: 45-minute-long trajectory obtained with normal INS filter operation (blue) in the beginning and the end, as well as with a (simulated) GPS outage of 30 minutes (red) corresponding to a distance traveled of 54.2 km. GPS positions are shown as black reference path.

Afterwards, the two trajectories coincide instantaneously, thus switching back to normal operation worked as expected.

When using the filter as a basis for UAS control, accurate orientation and airspeed are of highest importance—while position drift as assessed before does not matter for airplane stabilization. We show the orientation estimates with 3σ uncertainty bounds in regular operation in Figure 5.6.

In order to complete the presentation of estimated states, IMU biases, altitude and QFF, as well as wind estimates are provided in Figures 5.7, 5.8 and 5.9.

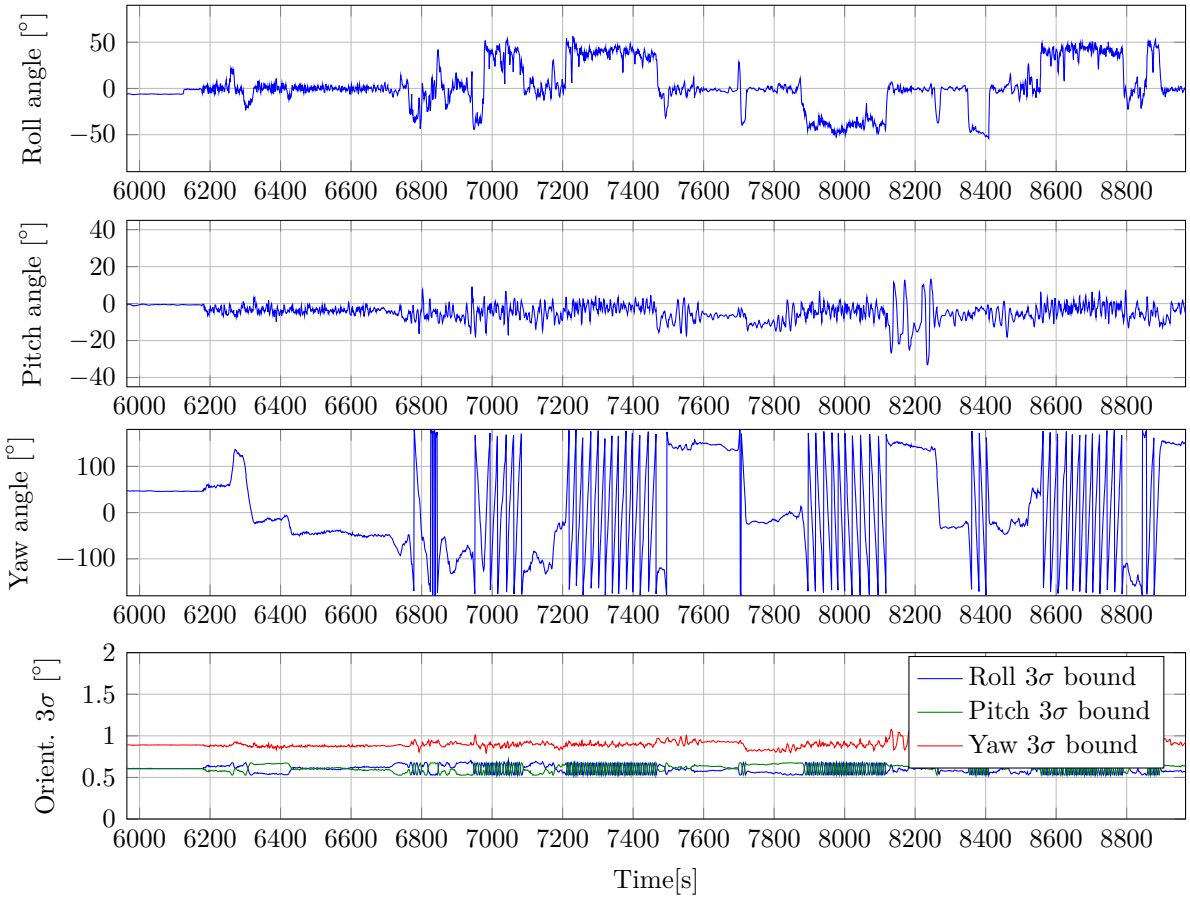


Figure 5.6: Orientation estimates and uncertainty bounds in regular operation.

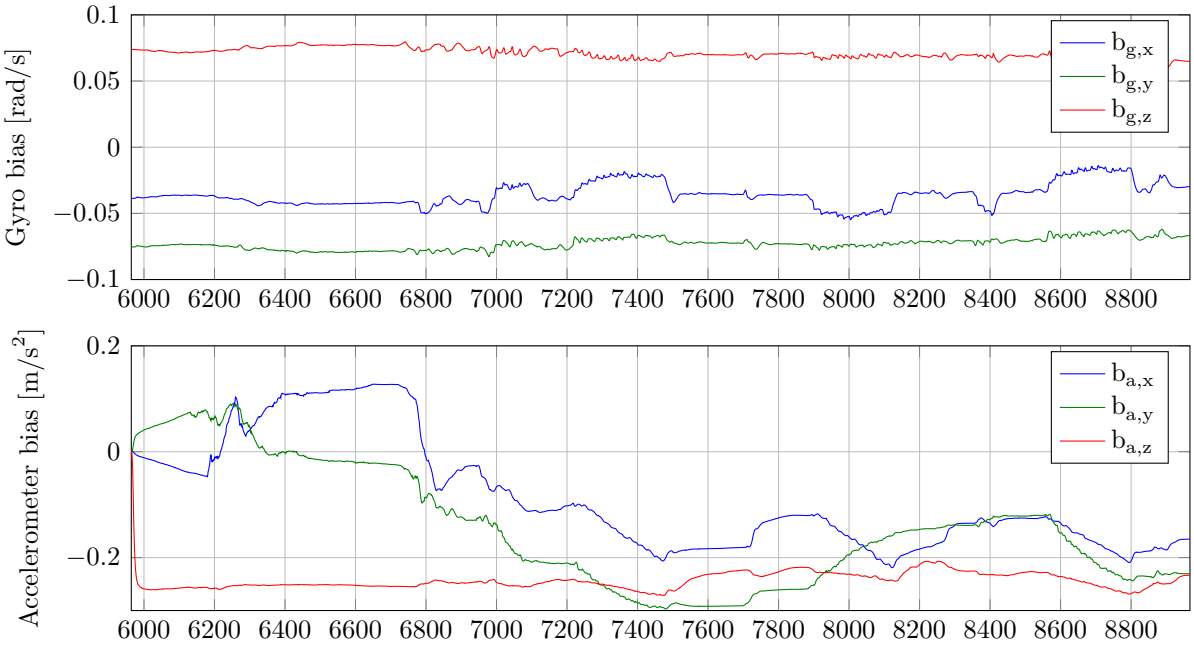


Figure 5.7: Accelerometer and gyro bias estimates in regular operation.

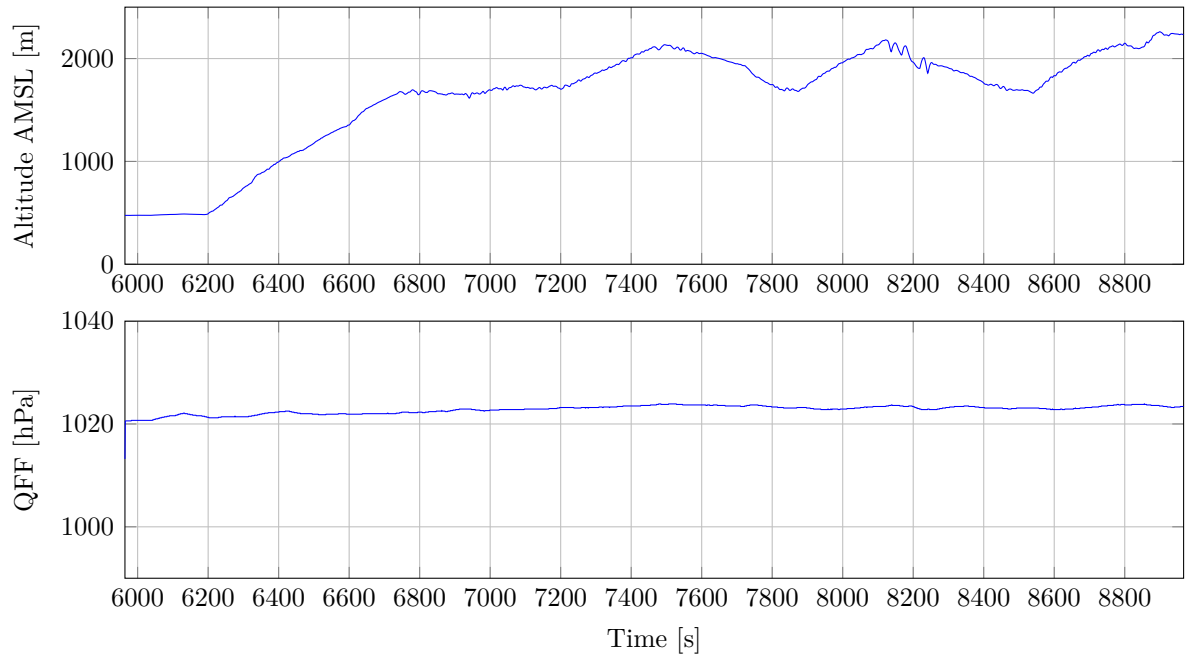


Figure 5.8: Estimates of altitude Above Mean Sea Level (AMSL) and QFF in regular operation.

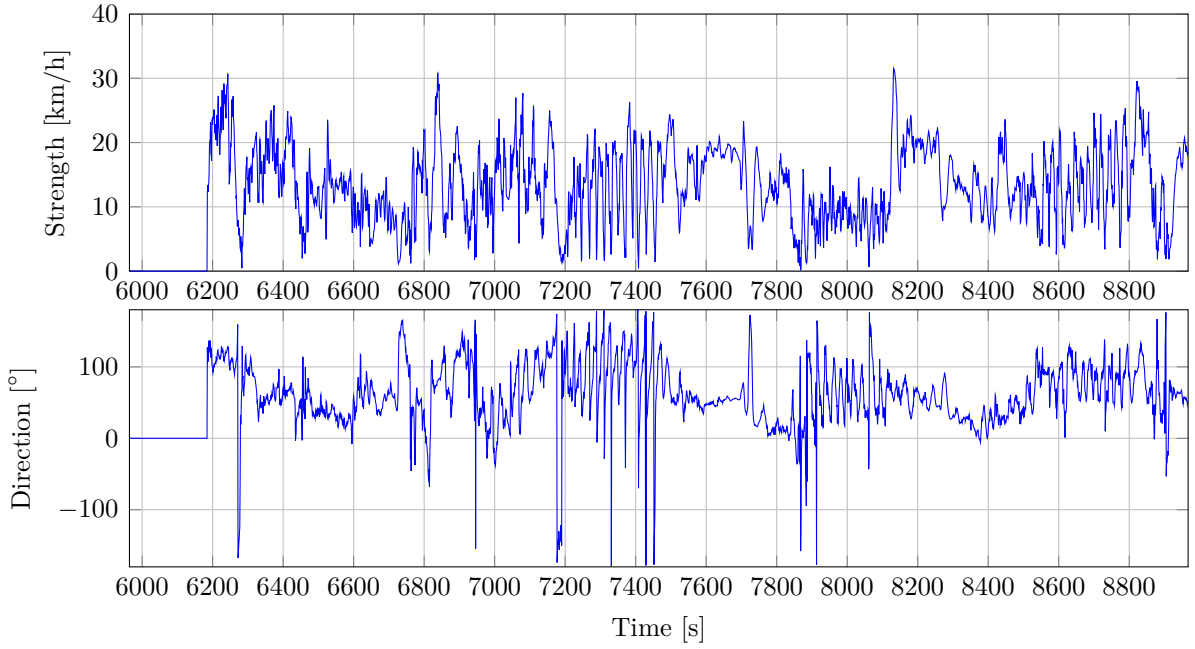


Figure 5.9: Wind estimate in regular operation.

Finally, we analyze the quantitative difference to these reference estimates when GPS is switched off. Figure 5.10 shows the orientation differences during the outage. The Root-Mean-Square (RMS) differences amount to 0.635°, 0.707°, and 0.952° for roll, pitch, and yaw angles, respectively. These deviations are typically acceptable for control. They are mainly caused by modeling uncertainty as well as measurement inaccuracy, in particular of pressure measurements at high sideslip angles.

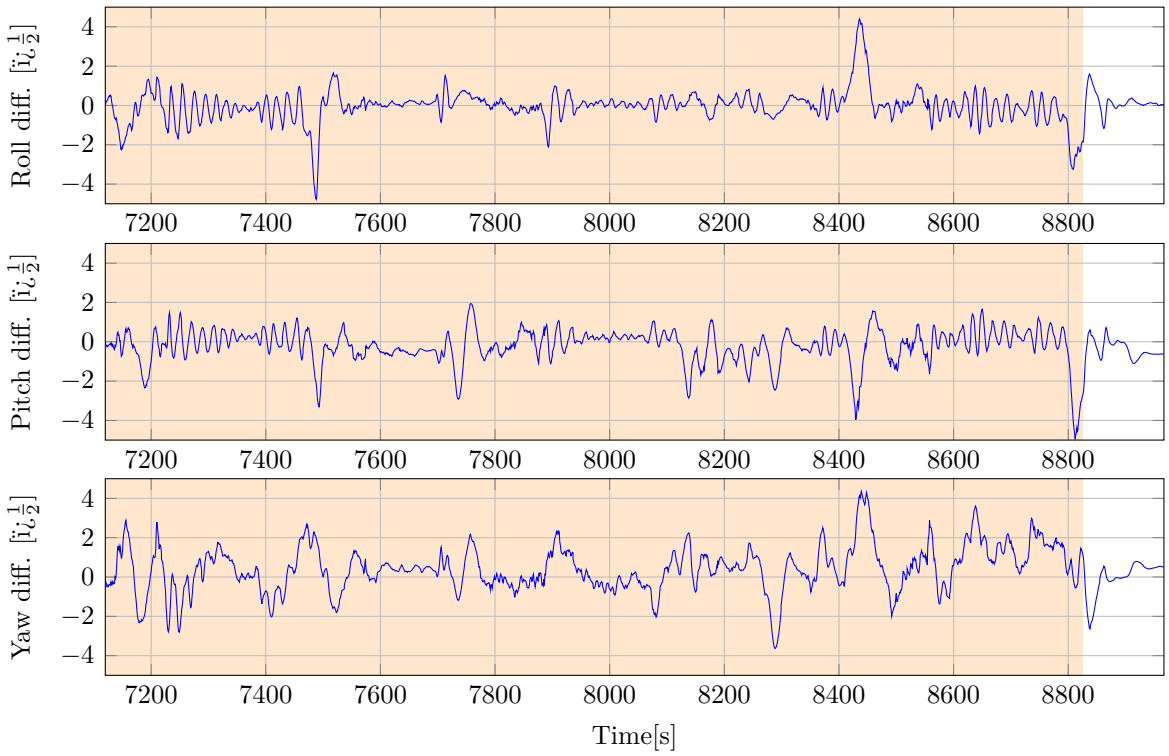


Figure 5.10: Differences of orientation estimates during simulated GPS outage (light orange background) as compared to regular operation with GPS.

5.5 Keyframe Visual-Inertial Odometry

After having shown how velocity, position, and single direction measurements can be used as updates to a light-weight inertial navigation filter, we now turn to fusing inertial measurements with visual measurements: due to the complementary characteristics of these sensing modalities, they have become a popular choice for accurate Simultaneous Localization and Mapping (SLAM) in mobile robotics.

While historically the problem has been addressed with filtering, advancements in visual estimation suggest that non-linear optimization offers superior accuracy, while still tractable in complexity thanks to the sparsity of the underlying problem. Taking inspiration from these findings, we formulate a rigorously probabilistic cost function that combines reprojection error of landmarks and inertial terms. We ensure real-time operation by limiting the optimization to a bounded window of keyframes by applying marginalization. Keyframes may be spaced in time by arbitrary intervals, while still related by linearized inertial terms. We present evaluation results on complementary datasets recorded with our custom-built stereo visual-inertial hardware that accurately synchronizes accelerometer and gyroscope measurements with imagery. A comparison of both a stereo and monocular version of our algorithm with and without online extrinsics estimation is shown with respect to ground truth. Furthermore, we compare the performance to an implementation of a state-of-the-art stochastic cloning sliding-window filter. This competitive reference implementation performs tightly-coupled filtering-based visual-inertial odometry. While our approach declaredly demands more computation, we show its superior performance in terms of accuracy.

5.5.1 Introduction

Visual and inertial measurements offer complementary properties which make them particularly suitable for fusion, in order to address robust and accurate localization and mapping, a primary need for any mobile robotic system. The rich representation of structure projected into an image, together with the accurate short-term estimates by gyroscopes and accelerometers contained in an IMU have been acknowledged to complement each other, with promising use-cases in airborne [Mourikis and Roumeliotis 2007; Weiss et al. 2012] and automotive [Li and Mourikis 2012b] navigation. Moreover, with the availability of these sensors in most smart phones, there is great interest and research activity in effective solutions to visual-inertial SLAM [Li et al. 2013].

Historically, there have been two main concepts towards approaching the visual-inertial estimation problem: batch *nonlinear optimization* methods and recursive *filtering* methods.While the former jointly minimizes the error originating from integrated IMU measurements and the (reprojection) errors from visual terms [Jung and Taylor 2001], recursive algorithms commonly use the IMU measurements for state propagation while updates originate from the visual observations [Chai et al. 2002; Roumeliotis et al. 2002].

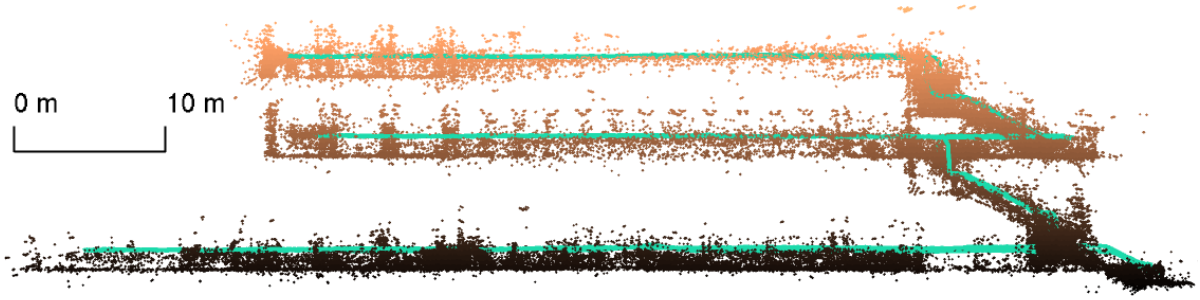
Batch approaches offer the advantage of repeated linearization of the inherently non-linear cost terms involved in the visual-inertial state estimation problem and thus they limit linearization errors. For a long time, however, the lack of computational resources made recursive algorithms a favorable choice for online estimation. Nevertheless, both paradigms have recently shown improvements over and compromises towards the other, so that recent work [Leutenegger et al. 2013; Nerurkar et al. 2013; Indelman et al. 2012] showed batch based algorithms reaching real-time operation and filtering based methods providing results of nearly equal quality [Mourikis and Roumeliotis 2007; Li et al. 2013] to batch-based methods. Leaving aside computational demands, batch based methods promise results of higher accuracy compared to filtering approaches, given the inherent algorithmic differences as discussed in detail later in this article.

Apart from the separation into batch and filtering, the visual-inertial fusion approaches found in the literature can be divided into two other categories: *loosely-coupled* systems independently estimate the pose by a vision only algorithm and fuse IMU measurements only in a separate estimation step, limiting computational complexity. *Tightly-coupled* approaches in contrast include both the measurements from the IMU and the camera into a common problem where all states are jointly estimated, thus considering all correlations amongst them. Comparisons of both approaches, however, show [Leutenegger et al. 2013] that these correlations are key for any high precision Visual-Inertial Navigation System (VINS), which is also why all high accuracy visual-inertial estimators presented recently have implemented a tightly coupled VINS: for example Mourikis and Roumeliotis [2007] proposed an Extended Kalman Filter (EKF) based real-time fusion using monocular vision, named Multi-State Constraint Kalman Filter (MSCKF). This work performs impressively with open loop errors below 0.5% of the distance traveled. We therefore compare our results to a competitive implementation of this sliding window filter with on-the-fly feature marginalization as published by Mourikis et al. [2009]. For simpler reference we denote this algorithm by "MSCKF" in the rest of the article, keeping in mind that the available reference implementation does not include *all* of the possible modifications from [Li and Mourikis 2012b,a; Li et al. 2013; Hesch et al. 2013].

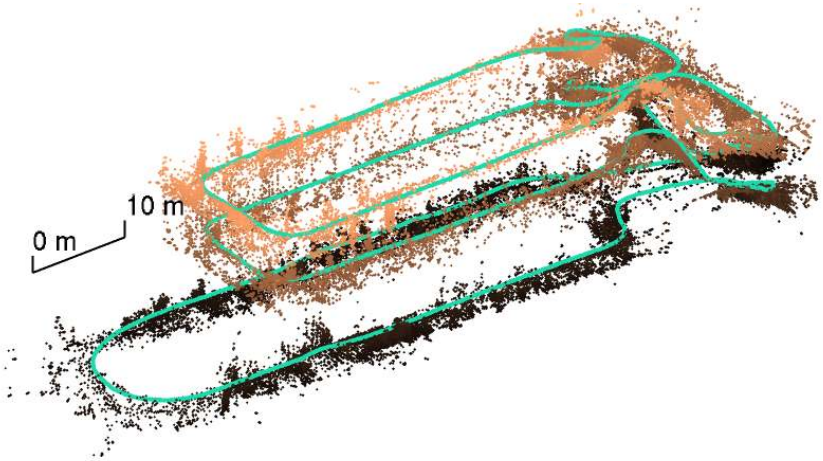
In this article, which extends our previous work [Leutenegger et al. 2013], we propose a method that respects the aforementioned findings: we advocate *tightly-coupled* fusion for

best exploitation of all measurements and *nonlinear optimization* where possible rather than filtering, in order to reduce suboptimality due to linearization. Furthermore, the optimization approach allows for employing robust cost functions which may drastically increase accuracy in the presence of outliers that may occasionally occur mostly in the visual part, even after application of sophisticated rejection schemes.

We devise a cost function that combines visual and inertial terms in a fully probabilistic manner. We adopt the concept of *keyframes* due to its successful application in classical vision-only approaches: it is implemented using partial linearization and marginalization, i.e. variable elimination—a compromise towards filtering that is made for real-time compliance and tractability. The keyframe paradigm accounts for drift-free estimation also when slow or no motion at all is present: rather than using an optimization window of time-successive poses, our kept keyframes may be spaced arbitrarily far in time, keeping visual constraints—while still incorporating an IMU term. Our formulation of relative uncertainty between keyframes takes inspiration from RSLAM [Mei et al. 2011], although our parameterization uses global coordinates. We provide a strictly probabilistic derivation of IMU error terms and the respective information matrix, relating successive image frames without explicitly introducing states at IMU-rate. At the system level, we developed both the hardware and the algorithms for accurate real-time SLAM, including robust keypoint matching, bootstrapping and outlier rejection using inertial cues.



(a) Side view of the ETH main building.



(b) 3D view of the building.

Figure 5.11: ETH main building indoor reconstruction of both structure and pose as resulting from our suggested visual-inertial odometry framework—i.e. without loop-closure (stereo variant in this case, including online camera extrinsics calibration). The stereo-vision plus IMU sensor was walked handheld for 470 m through loops on three floors as well as through staircases.

Figure 5.11 shows the output of our stereo visual-inertial odometry algorithm as run on an indoor dataset: the stereo-vision plus IMU sensor was walked for 470 m through several floors and staircases in the ETH main building. Along with the state consisting of pose, speed, and IMU biases, we also obtain an impression of the environment represented as a sparse map of 3D landmarks. Note that the map and path are automatically aligned with gravity thanks to tightly coupled IMU fusion.

After having shown the superior performance of the suggested method compared to a loosely-coupled approach, we present extensive evaluation results with respect to a stochastic cloning sliding window filter (following the MSCKF implementation of Mourikis et al. [2009]) in terms of accuracy on different motion profiles. Various datasets were recorded with our newest hardware for evaluation, featuring individual characteristics in terms of motion, appearance, and scene depth ranging from hand-held indoor motion to bicycle riding. Our algorithm consistently outperforms the filtering-based method, while it admittedly incurs higher computational cost. In the following, we will present a generic N-camera ($N \geq 1$) visual-inertial odometry framework. In the stereo-version, the performance will gradually transform into the monocular case when the ratio between camera baseline and distance to structure becomes small. We furthermore present the formulation and results for online camera extrinsics estimation that may be applied after standard intrinsics calibration. Evaluation results demonstrate the applicability of this method, when initializing with inaccurate camera pose estimates with respect to the IMU.

The remainder of this Section is structured as follows: Section 5.5.2 introduces the visual-inertial sensor setup with related definitions. The nonlinear error terms from camera and IMU measurements are described in-depth in Section 5.5.3, which is then followed by an overview of frontend processing and initialization in Section 5.5.4. As a last key element of the method, Section 5.5.5 introduces how the keyframe concept is applied by marginalization. Section 5.5.6 describes the experimental setup, evaluation scheme and presents extensive results on the different datasets.

5.5.2 Visual-Inertial Sensor Setup, Notation and States

The performance of the proposed method is evaluated using an IMU and camera setup schematically depicted in Figure 5.12. It is used both in monocular and stereo mode,

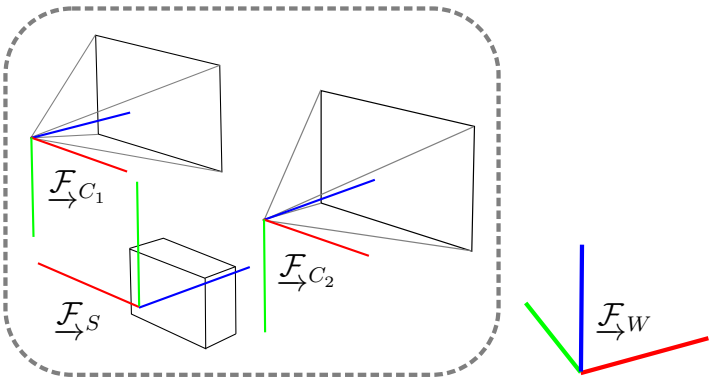


Figure 5.12: Coordinate frames involved in the hardware setup used: two cameras are placed as a stereo setup with respective frames, $\underrightarrow{\mathcal{F}}_{C_i}, i \in \{1,2\}$. IMU data is acquired in $\underrightarrow{\mathcal{F}}_{S}$. The algorithms estimate the position and orientation of $\underrightarrow{\mathcal{F}}_{S}$ with respect to the world (inertial) frame $\underrightarrow{\mathcal{F}}_{W}$.

where we want to emphasize that our methodology is generic enough to handle an N-camera setup. Inside the tracked body that is represented relative to an inertial frame, $\underrightarrow{\mathcal{F}}_W$, we distinguish camera frames, $\underrightarrow{\mathcal{F}}_{C_i}$ (subscipted with $i = 1, \dots N$), and the IMU-sensor frame, $\underrightarrow{\mathcal{F}}_S$.

The variables to be estimated comprise the robot states at the image times (index k) $\mathbf{x}_{\mathrm{R}}^k$ and landmarks $\mathbf{x}_{\mathrm{L}}$. $\mathbf{x}_{\mathrm{R}}$ holds the robot position in the inertial frame ${}_W\mathbf{r}_S$, the body orientation quaternion $\mathbf{q}_{WS}$, the velocity expressed in the sensor frame ${}_S\mathbf{v}_{WS}$ (written in short as ${}_S\mathbf{v}$), as well as the biases of the gyroscopes $\mathbf{b}_{\mathrm{g}}$ and the biases of the accelerometers $\mathbf{b}_{\mathrm{a}}$. Thus, $\mathbf{x}_{\mathrm{R}}$ is written as:

$$\mathbf{x}_{\mathrm{R}} := \left[{}_W\mathbf{r}_S{}^T, \mathbf{q}_{WS}^T, {}_S\mathbf{v}^T, \mathbf{b}_{\mathrm{g}}^T, \mathbf{b}_{\mathrm{a}}^T\right]^T \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^9. \tag{5.49}$$

Furthermore, we use a partition into the pose states $\mathbf{x}_{\mathrm{T}} := [{}_W\mathbf{r}_S{}^T, \mathbf{q}_{WS}^T]^T$ and the speed/bias states $\mathbf{x}_{\mathrm{sb}} := [{}_S\mathbf{v}^T, \mathbf{b}_{\mathrm{g}}^T, \mathbf{b}_{\mathrm{a}}^T]^T$. Optionally, we may include camera extrinsics estimation as part of an online calibration process. Camera extrinsics denoted $\mathbf{x}_{C_i} := [{}_S\mathbf{r}_{C_i}{}^T, \mathbf{q}_{SC_i}^T]^T$ can either be treated as constant entities to be calibrated or time-varying states subjected to a first-order Gaussian process allowing to track changes that may occur e.g. due to temperature-induced mechanical deformation of the setup.

We obtain the minimal robot error state vector

$$\delta\boldsymbol{\chi}_{\mathrm{R}} = \left[\delta\mathbf{p}^T, \delta\boldsymbol{\alpha}^T, \delta\mathbf{v}^T, \delta\mathbf{b}_{\mathrm{g}}^T, \delta\mathbf{b}_{\mathrm{a}}^T\right]^T \in \mathbb{R}^{15}. \tag{5.50}$$

Analogously to the robot state decomposition $\mathbf{x}_{\mathrm{T}}$ and $\mathbf{x}_{\mathrm{sb}}$, we use the pose error state $\delta\boldsymbol{\chi}_{\mathrm{T}} := [\delta\mathbf{p}^T, \delta\boldsymbol{\alpha}^T]^T$ and the speed/bias error state $\delta\boldsymbol{\chi}_{\mathrm{sb}} := [\delta\mathbf{v}^T, \delta\mathbf{b}_{\mathrm{g}}^T, \delta\mathbf{b}_{\mathrm{a}}^T]^T$.

The j^{th} landmark is represented in homogeneous (World) coordinates: $\mathbf{x}_{\mathrm{L}^j} := {}_W\boldsymbol{l}^j \in \mathbb{R}^4$. At this point, we set the fourth component to one. As landmark perturbation, we use the simple Euclidean version $\delta\boldsymbol{\beta} \in \mathbb{R}^3$.

5.5.3 Batch Visual SLAM with Inertial Terms

In visual odometry and SLAM, a nonlinear optimization is formulated to find the camera poses and landmark positions by minimizing the reprojection error of landmarks observed in camera frames. Figure 5.13 shows the respective graph representation inspired by

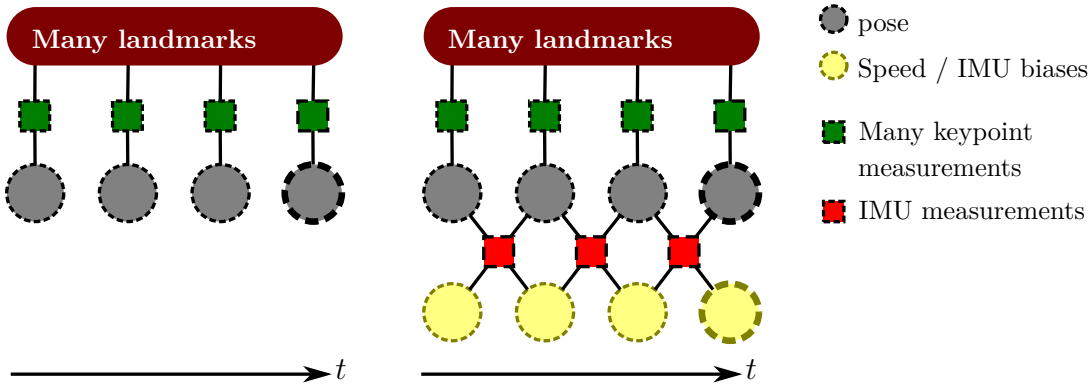


Figure 5.13: Graphs of the state variables and measurements involved in the visual SLAM problem (left) versus visual-inertial SLAM (right): incorporating inertial measurements introduces temporal constraints, and necessitates a state augmentation by the robot speed as well as IMU biases.

[Thrun and Montemerlo 2006]: it displays measurements as edges with square boxes and estimated quantities as round nodes. As soon as inertial measurements are introduced, they not only create temporal constraints between successive poses, but also between successive speed and IMU bias estimates of both accelerometers and gyroscopes by which the robot state vector is augmented. In this section, we present our approach of incorporating inertial measurements into batch visual SLAM.

We seek to formulate the visual-inertial localization and mapping problem as one joint optimization of a cost function $J(\mathbf{x})$ containing both the (weighted) reprojection errors $\mathbf{e}_\mathrm{r}$ and the (weighted) temporal error term from the IMU $\mathbf{e}_\mathrm{s}$:

$$J(\mathbf{x}) := \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{j \in \mathcal{J}(i,k)} \mathbf{e}_\mathrm{r}^{i,j,k\,T} \mathbf{W}_\mathrm{r}^{i,j,k} \mathbf{e}_\mathrm{r}^{i,j,k} + \sum_{k=1}^{K-1} \mathbf{e}_\mathrm{s}^{k\,T} \mathbf{W}_\mathrm{s}^{k} \mathbf{e}_\mathrm{s}^{k}, \tag{5.51}$$

where i is the camera index of the assembly, k denotes the camera frame index, and j denotes the landmark index. The indices of landmarks visible in the k^{th} frame and the i^{th} camera are written as the set $\mathcal{J}(i,k)$. Furthermore, $\mathbf{W}_\mathrm{r}^{i,j,k}$ represents the information matrix of the respective landmark measurement, and $\mathbf{W}_\mathrm{s}^{k}$ the information of the k^{th} IMU error.

In the following, we will present the reprojection error formulation. Afterwards, an overview on IMU kinematics combined with bias term modeling is given, upon which we base the IMU error term.

Reprojection Error Formulation

We use a rather standard formulation of the reprojection error adapted with minor modifications from Furgale [2011]:

$$\mathbf{e}_\mathrm{r}^{i,j,k} = \mathbf{z}^{i,j,k} - \mathbf{h}_i \left(\boldsymbol{T}_{CiS}^{k}\, \boldsymbol{T}_{SW}^{k}\, {}_W\boldsymbol{l}^{j} \right). \tag{5.52}$$

Hereby $\mathbf{h}_i(\cdot)$ denotes the camera projection model (which may include distortion) and $\mathbf{z}^{i,j,k}$ stands for the measurement image coordinates. We also provide the Jacobians here, since they are not only needed for efficient solving, but also play a central role in the marginalization step:

$$\frac{\partial \mathbf{e}_\mathrm{r}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_\mathrm{T}{}^{k}} = \mathbf{J}_{\mathrm{r},i} \overline{\boldsymbol{T}}_{C_iS}^{k} \begin{bmatrix} \overline{\mathbf{C}}_{SW}^{k}\, {}_W\overline{\boldsymbol{l}}_4^{j} & \overline{\mathbf{C}}_{SW}^{k} \left[{}_W\overline{\boldsymbol{l}}_{1:3}^{j} - {}_W\mathbf{r}_{SW}^{k} \overline{\boldsymbol{l}}_4^{j} \right]^\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}, \tag{5.53}$$

$$\frac{\partial \mathbf{e}_\mathrm{r}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_\mathrm{L}{}^{j}} = -\mathbf{J}_{\mathrm{r},i} \overline{\boldsymbol{T}}_{C_iS}^{k} \begin{bmatrix} \overline{\mathbf{C}}_{SW}^{k} \\ \mathbf{0}_{1\times 3} \end{bmatrix}, \tag{5.54}$$

$$\frac{\partial \mathbf{e}_\mathrm{r}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_\mathrm{Ci}{}^{k}} = \mathbf{J}_{\mathrm{r},i} \begin{bmatrix} \mathbf{C}_{C_iS}^{k}\, {}_S\overline{\boldsymbol{l}}_4^{j} & \mathbf{C}_{C_iS}^{k} \left[{}_S\overline{\boldsymbol{l}}_{1:3}^{j} - {}_S\mathbf{r}_{C_iS}^{k} \overline{\boldsymbol{l}}_4^{j} \right]^\times \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix}, \tag{5.55}$$

where $\mathbf{J}_{\mathrm{r},i}$ denotes the Jacobian matrix of the projection $\mathbf{h}_i(\cdot)$ into the i^{th} camera (including distortion) with respect to a landmark in homogeneous coordinates, and variables with an overbar represent our current guess. Our framework currently supports radial-tangential as well as equidistant distortion models.

Formulation of the IMU Measurement Error Term

Figure 5.14 illustrates the difference in measurement rates with camera measurements taken at time steps k and $k+1$, as well as faster IMU-measurements that are not necessarily synchronized with the camera measurements. Note also the introduction of a

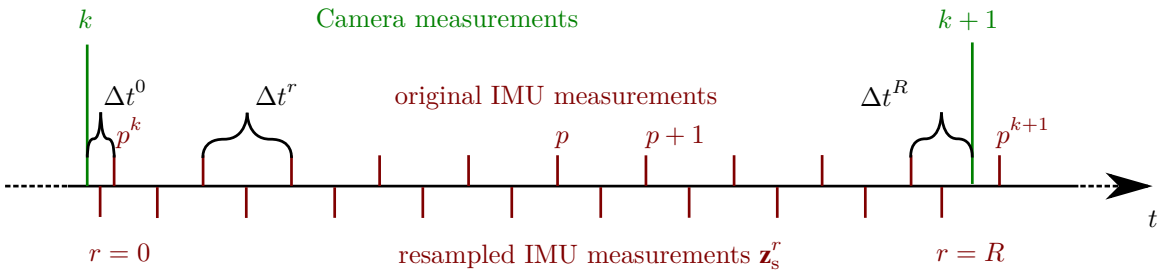


Figure 5.14: Different rates of IMU and camera: one IMU term uses all accelerometer and gyro readings between successive camera measurements.

local time index $r = 0, \ldots, R$ between camera measurements, along with respective time increments Δt^r.

We need the IMU error term $\mathbf{e}_{\mathrm{s}}^k(\mathbf{x}_{\mathrm{R}}^k, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^k)$ to be a function of robot states at steps k and $k+1$ as well as of all the IMU measurements in-between these time instances (comprising accelerometer and gyro readings) summarized as $\mathbf{z}_{\mathrm{s}}^k$. Hereby we have to assume an approximate normal conditional probability density f for given robot states at camera measurements k and $k+1$:

$$f\left(\mathbf{e}_{\mathrm{s}}^k | \mathbf{x}_{\mathrm{R}}^k, \mathbf{x}_{\mathrm{R}}^{k+1}\right) \approx \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{\mathrm{s}}^k\right). \tag{5.56}$$

We are employing the propagation equations

$$\begin{aligned}
{}_W\dot{\mathbf{r}}_S &= \mathbf{C}_{WS}\, {}_S\mathbf{v}\,, \\
\dot{\mathbf{q}}_{WS} &= \frac{1}{2}\boldsymbol{\Omega}\left({}_S\boldsymbol{\omega}\right)\mathbf{q}_{WS}\,, \\
{}_S\dot{\mathbf{v}} &= {}_S\tilde{\mathbf{a}} + \mathbf{w}_{c,\mathrm{a}} - \mathbf{b}_{\mathrm{a}} + \mathbf{C}_{SW}\, {}_W\mathbf{g} - \left({}_S\boldsymbol{\omega}\right) \times {}_S\mathbf{v}\,, \\
\dot{\mathbf{b}}_{\mathrm{g}} &= \mathbf{w}_{c,b_{\mathrm{g}}}, \\
\dot{\mathbf{b}}_{\mathrm{a}} &= -\frac{1}{\tau}\mathbf{b}_{\mathrm{a}} + \mathbf{w}_{c,b_{\mathrm{a}}},
\end{aligned} \tag{5.57}$$

where the matrix $\boldsymbol{\Omega}$ is formed from the estimated angular rate ${}_S\boldsymbol{\omega} = {}_S\tilde{\boldsymbol{\omega}} + \mathbf{w}_{c,\mathrm{g}} - \mathbf{b}_{\mathrm{g}}$, with gyro measurement ${}_S\tilde{\boldsymbol{\omega}}$:

$$\boldsymbol{\Omega}\left({}_S\boldsymbol{\omega}\right) := \begin{bmatrix} -{}_S\boldsymbol{\omega} \\ 0 \end{bmatrix}^{\oplus}. \tag{5.58}$$

Furthermore, the linearized error state dynamics as detailed in Section 5.3 are employed in order to formulate a prediction $\hat{\mathbf{x}}_{\mathrm{R}}^{k+1}\left(\mathbf{x}_{\mathrm{R}}^k, \mathbf{z}_{\mathrm{s}}^k\right)$ with associated conditional covariance $\mathbf{P}\left(\delta\hat{\mathbf{x}}_{\mathrm{R}}^{k+1} | \mathbf{x}_{\mathrm{R}}^k, \mathbf{z}_{\mathrm{s}}^k\right)$. The respective computation requires numeric integration; as common in related literature [Mourikis and Roumeliotis 2007], we applied the classical Runge-Kutta method, in order to obtain discrete time nonlinear state transition equations $\mathbf{f}_{\mathrm{d}}(\bar{\mathbf{x}}_{\mathrm{R}}^k)$ and the error state transition matrix $\mathbf{F}_{\mathrm{d}}(\bar{\mathbf{x}}_{\mathrm{R}}^k)$. The latter is found by integrating $\delta\dot{\boldsymbol{\chi}}_{\mathrm{R}} = \mathbf{F}_c(\bar{\mathbf{x}}_{\mathrm{R}})\delta\boldsymbol{\chi}_{\mathrm{R}}$ over Δt^r keeping $\delta\boldsymbol{\chi}_{\mathrm{R}}$ symbolic.

Using the prediction, we can now formulate the IMU error term as:

$$\mathbf{e}_{\mathrm{s}}^{k}\left(\mathbf{x}_{\mathrm{R}}^{k}, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^{k}\right) = \begin{bmatrix} {}_{W}\hat{\mathbf{r}}_{S}^{k+1} - {}_{W}\mathbf{r}_{S}^{k+1} \\ 2\left[\hat{\mathbf{q}}_{WS}^{k+1} \otimes \mathbf{q}_{WS}^{k+1\,-1}\right]_{1:3} \\ \hat{\mathbf{x}}_{\mathrm{sb}}^{k+1} - \mathbf{x}_{\mathrm{sb}}^{k+1} \end{bmatrix} \in \mathbb{R}^{15}. \tag{5.59}$$

This is simply the difference between the prediction based on the previous state and the actual state—except for orientation, where we use a simple multiplicative minimal error.

Next, upon application of the error propagation law, the associated information matrix $\mathbf{W}_{\mathrm{s}}^{k}$ is found as:

$$\mathbf{W}_{\mathrm{s}}^{k} = \mathbf{R}_{\mathrm{s}}^{k\,-1} = \left(\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} \mathbf{P}\left(\delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1} | \mathbf{x}_{\mathrm{R}}^{k}, \mathbf{z}_{\mathrm{s}}^{k}\right) \frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}}^{T}\right)^{-1}. \tag{5.60}$$

The Jacobian $\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}}$ is straightforward to obtain but non-trivial, since the orientation error will be nonzero in general:

$$\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} = \begin{bmatrix} \mathbf{1}_{3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{3\times3} & \left[\hat{\mathbf{q}}_{WS}^{k+1} \otimes \mathbf{q}_{WS}^{k+1\,-1}\right]_{1:3,1:3}^{\oplus} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{1}_{9} \end{bmatrix}. \tag{5.61}$$

Finally, the Jacobians with respect to $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k}$ and $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k+1}$ will be needed for efficient solution of the optimization problem. While differentiating with respect to $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k+1}$ is straightforward (but non-trivial), some attention is given to the other Jacobian. Recall that the IMU error term (5.59) is calculated by iteratively applying the prediction. Differentiation with respect to the state $\delta\boldsymbol{\chi}_{\mathrm{R}}^{k}$ thus leads to application of the chain rule, yielding

$$\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\boldsymbol{\chi}_{\mathrm{R}}^{k}} = \frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta\hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} \mathbf{F}_{\mathrm{d}}(\bar{\hat{\mathbf{x}}}_{\mathrm{R}}^{R}, \Delta t^{R}) \mathbf{F}_{\mathrm{d}}(\bar{\hat{\mathbf{x}}}_{\mathrm{R}}^{R-1}, \Delta t^{R-1}) \ldots \mathbf{F}_{\mathrm{d}}(\bar{\hat{\mathbf{x}}}_{\mathrm{R}}^{1}, \Delta t^{1}) \mathbf{F}_{\mathrm{d}}(\bar{\mathbf{x}}_{\mathrm{R}}^{k}, \Delta t^{0}). \tag{5.62}$$

5.5.4 Frontend Overview

This section overviews the image processing steps and data association along with outlier detection and initialization of landmarks and states.

Keypoint Detection, Matching, and Variable Initialization

Our processing pipeline employs a customized multi-scale SSE-optimized Harris corner detector [Harris and Stephens 1988] followed by BRISK descriptor extraction [Leutenegger et al. 2011]. The detection scheme favors a uniform keypoint distribution in the image by gradually suppressing corners with weaker corner response close to a stronger corner. BRISK would allow automatic orientation detection—however, better matching results are obtained by extracting descriptors oriented along the gravity direction that is projected into the image. This direction is globally observable thanks to IMU fusion.

As a first step to initialization and matching, we propagate the last pose using acquired IMU measurements in order to obtain a preliminary uncertain estimate of the states.

Assume a set of past frames (including keyframes) as well as a local map consisting of landmarks with sufficiently well known 3D position is available at this point (see 5.5.4 for details). As a first stage of establishing correspondences, we perform a 3D-2D matching step. Given the current pose prediction, all landmarks that should be visible are considered for brute-force descriptor matching. Outliers are only rejected afterwards. This scheme may seem illogical to the reader who might intuitively want to apply the inverse order in the sense of a guided matching strategy; however, owing to the super-fast matching of binary descriptors, it would actually be more expensive to first look at image-space consistency. The outlier rejection consists of two steps: first of all, we use the uncertain pose predictions in order to perform a Mahalanobis test in image coordinates. Second, an absolute pose RANSAC provided in OpenGV [Kneip and Furgale 2014] is applied.

Next, 2D-2D matching is performed in order to associate keypoints without 3D landmark correspondences. Again, we use brute-force matching first, followed by triangulation, in order to initialize landmark positions and as a first step to rejecting outlier pairings. Both stereo-triangulation across stereo image pairs (in the non-mono case) is performed, as well as between the current frame and any previous frame available. Only triangulations with sufficiently low depth uncertainty are labeled to be initialized—the rest will be treated as 2D measurements in subsequent matching. Finally, a relative RANSAC step [Kneip and Furgale 2014] is performed between the current frame and the newest keyframe. The respective pose guess is furthermore used for bootstrapping in the very beginning. Figure 5.15 illustrates a typical detection and matching result in the stereo case. Note the challenging illumination with overexposed sky.

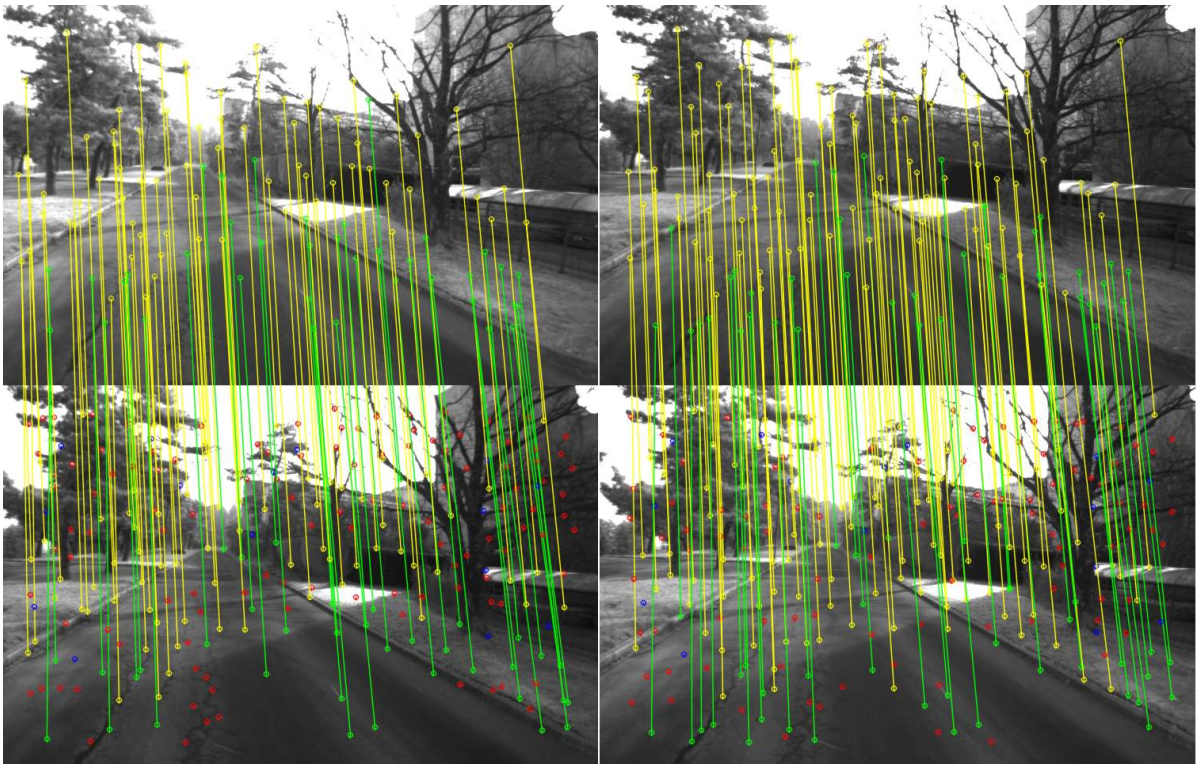


Figure 5.15: Visualization of typical data association on a bicycle dataset: current stereo image pair (bottom) with match lines to the newest keyframe (top). Green stands for a 3D-2D match, yellow for 2D-2D match, blue for keypoints with left-right stereo match only, and red keypoints are left unmatched.

Keyframe Selection

For the subsequent optimization, a bounded set of camera frames is maintained, i.e. poses with associated image(s) taken at that time instant; all landmarks co-visible in these images are kept in the local map. As illustrated in Figure 5.16, we distinguish two kinds of frames: we introduce a temporal window of the S most recent frames including the

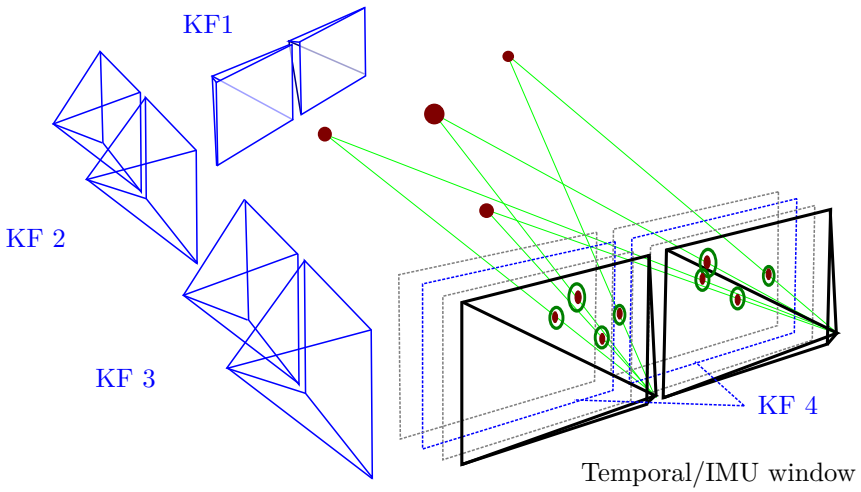


Figure 5.16: Frames kept for matching and subsequent optimization in the stereo case: in this example, $M = 3$ keyframes and $S = 4$ most current frames are used.

current frame; and we use a number of M keyframes that may have been taken far in the past. For keyframe selection, we use a simple heuristic: if the hull of projected and matched landmarks covers less than some percentage of the image (we use around 50%), or if the ratio of matched versus detected keypoints is small (below around 20%), the frame is inserted as keyframe.

5.5.5 Keyframes and Marginalization

In contrast to the vision-only case, it is not obvious how nonlinear temporal constraints from the IMU can reside in a bounded optimization window containing keyframes that may be arbitrarily far spaced in time. In the following, we first provide the mathematical foundations for marginalization, i.e. elimination of states in nonlinear optimization, and apply them to visual-inertial odometry.

Formulation of Marginalization in Nonlinear Optimization

A Gauss-Newton system of equations is constructed from all the error terms, Jacobians and information matrices, taking the form $\mathbf{H}\delta\boldsymbol{\chi} = \mathbf{b}$. Let us consider a set of states to be marginalized out, $\mathbf{x}_\mu$, the set of all states related to those by error terms, $\mathbf{x}_\lambda$, and the set of remaining states, $\mathbf{x}_\rho$. Due to conditional independence, we can simplify the marginalization step and only apply it to a sub-problem:

$$\begin{bmatrix} \mathbf{H}_{\mu\mu} & \mathbf{H}_{\mu\lambda_1} \\ \mathbf{H}_{\lambda_1\mu} & \mathbf{H}_{\lambda_1\lambda_1} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\chi}_\mu \\ \delta\boldsymbol{\chi}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\mu \\ \mathbf{b}_{\lambda_1} \end{bmatrix} \tag{5.63}$$

Application of the Schur complement operation yields:

$$\mathbf{H}^*_{\lambda_1\lambda_1} := \mathbf{H}_{\lambda_1\lambda_1} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{H}_{\mu\lambda_1}, \tag{5.64a}$$

$$\mathbf{b}^*_{\lambda_1} := \mathbf{b}_{\lambda_1} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{b}_{\mu}, \tag{5.64b}$$

where $\mathbf{b}^*_{\lambda_1}$ and $\mathbf{H}^*_{\lambda_1\lambda_1}$ are nonlinear functions of $\mathbf{x}_\lambda$ and $\mathbf{x}_\mu$.

The equations in (5.64) describe a single step of marginalization. In our keyframe-based approach, we must apply the marginalization step repeatedly and incorporate the resulting information as a prior in our optimization while our state estimate continues to change. Hence, we fix the linearization point around $\mathbf{x}_0$, the value of $\mathbf{x}$ at the time of marginalization. The finite deviation $\Delta\boldsymbol{\chi} := \Phi^{-1}(\log(\overline{\mathbf{x}} \boxplus \mathbf{x}_0^{-1})))$ represents state updates that occur after marginalization, where $\overline{\mathbf{x}}$ is our current estimate for $\mathbf{x}$. In other words, $\mathbf{x}$ is composed as

$$\mathbf{x} = \exp\left(\Phi(\delta\boldsymbol{\chi})\right) \boxplus \underbrace{\exp\left(\Phi(\Delta\boldsymbol{\chi})\right) \boxplus \mathbf{x}_0}_{=\overline{\mathbf{x}}}. \tag{5.65}$$

This generic formulation allows us to apply prior information on minimal coordinates to any of our state variables—including unit length quaternions. Introducing $\Delta\boldsymbol{\chi}$ allows the right hand side to be approximated (to first order) as

$$\mathbf{b} + \left.\frac{\partial \mathbf{b}}{\partial \Delta\boldsymbol{\chi}}\right|_{\mathbf{x}_0} \Delta\boldsymbol{\chi} = \mathbf{b} - \mathbf{H}\Delta\boldsymbol{\chi}. \tag{5.66}$$

Again using the partition into $\delta\boldsymbol{\chi}_\mu$ and $\delta\boldsymbol{\chi}_\lambda$, we can now write (5.66) as the right-hand side of the Gauss-Newton system (5.64b) as:

$$\begin{bmatrix} \mathbf{b}_\mu \\ \mathbf{b}_{\lambda_1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{\mu,0} \\ \mathbf{b}_{\lambda_1,0} \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{\mu\mu} & \mathbf{H}_{\mu\lambda_1} \\ \mathbf{H}_{\lambda_1\mu} & \mathbf{H}_{\lambda_1\lambda_1} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\chi}_\mu \\ \Delta\boldsymbol{\chi}_\lambda \end{bmatrix}. \tag{5.67}$$

In this form, i.e. plugging in (5.67), the right-hand side (5.64) becomes

$$\mathbf{b}^*_{\lambda_1} = \underbrace{\mathbf{b}_{\lambda_1,0} - \mathbf{H}_{\lambda_1\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{b}_{\mu,0}}_{\mathbf{b}^*_{\lambda_1,0}} - \mathbf{H}^*_{\lambda_1\lambda_1}\Delta\boldsymbol{\chi}_{\lambda_1}. \tag{5.68}$$

The marginalization procedure thus consists of applying (5.64a) and (5.68).

In the case where marginalized nodes comprise landmarks at infinity (or sufficiently close to infinity), or landmarks visible only in one camera from a single pose, the Hessian blocks associated with those landmarks will be (numerically) rank-deficient. We thus employ the pseudo-inverse $\mathbf{H}^+_{\mu\mu}$, which provides a solution for $\delta\boldsymbol{\chi}_\mu$ given $\delta\boldsymbol{\chi}_\lambda$ with a zero-component into null space direction.

The formulation described above introduces a fixed linearization point for both the states that are marginalized $\mathbf{x}_\mu$, as well as the remaining states $\mathbf{x}_\lambda$. This will also be used as as point of reference for *all future* linearizations of terms involving these states: this procedure is referred to as using "first estimate Jacobians" and was applied in [Dong-Si and Mourikis 2011], with the aim of minimizing erroneous accumulation of information. After application of (5.64), we can remove the nonlinear terms consumed and add the marginalized $\mathbf{H}^*_{\lambda_1\lambda_1}$ and $\mathbf{b}^*_{\lambda_1}$ as summands to construct the overall Gauss-Newton system. The contribution to the chi-square error may be written as $\chi^2_{\lambda_1} = \mathbf{b}^{*T}_{\lambda_1}\mathbf{H}^{*+}_{\lambda_1\lambda_1}\mathbf{b}^*_{\lambda_1}$.

Marginalization Applied to Keyframe-Based Visual-Inertial SLAM

The initially marginalized error term is constructed from the first $M+1$ frames $\mathbf{x}_{\mathrm{T}}^{k}, k = 1, \ldots, M+1$ with respective speed and bias states as visualized graphically in Figure 5.17. The M first frames will all be interpreted as keyframes and the marginalization step consists of eliminating the corresponding speed and bias states. Note that before

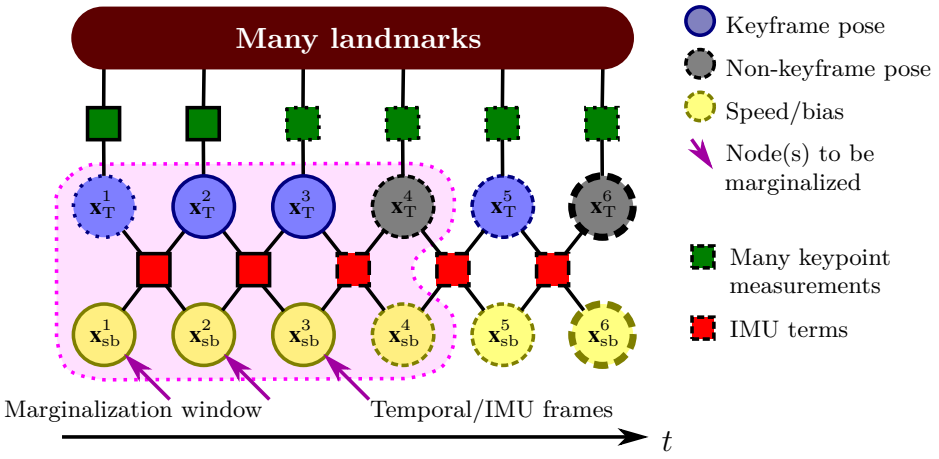


Figure 5.17: Graph showing the initial marginalization on the first $M+1$ frames: speed and bias states outside the temporal window of size $S = 3$ are marginalized out.

marginalization, we transform all error terms relating variables to be marginalized into one linear error term according to (5.67), which will persist and form a part of any subsequent marginalization step.

When a new frame $\mathbf{x}_{\mathrm{T}}^{c}$ (current frame, index c) is inserted into the optimization window, we apply a marginalization operation. In the case where the oldest frame in the temporal window ($\mathbf{x}_{\mathrm{T}}^{c-S}$) is not a keyframe, we will drop all its landmark measurements and then marginalize it out together with the oldest speed and bias states. In other words, all states are marginalized out, but no landmarks. Figure 5.18 illustrates this process. Dropping landmark measurements is suboptimal; however, it keeps the problem

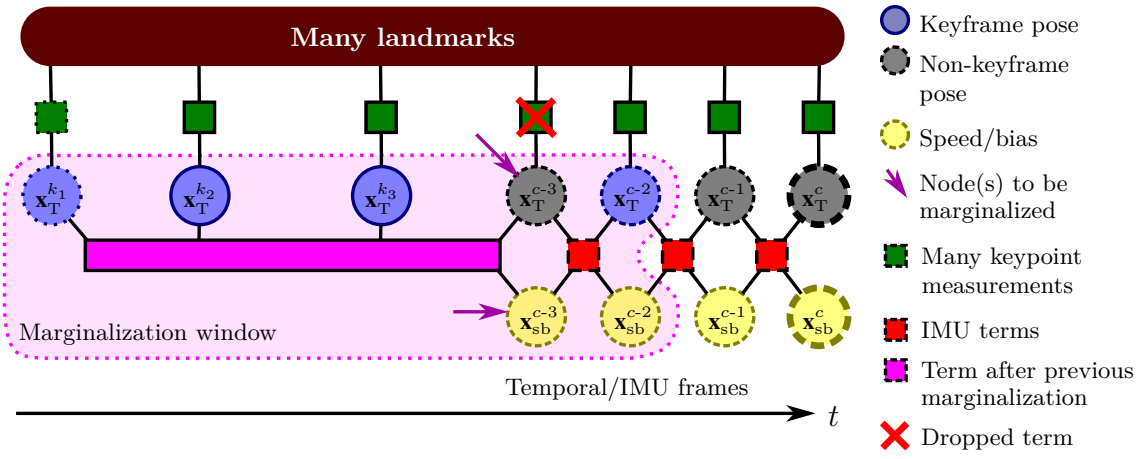


Figure 5.18: Graph illustration with $M = 3$ keyframes and an IMU/temporal node size $S = 3$. A regular frame is slipping out of the temporal window. All corresponding keypoint measurements are dropped and the pose as well as speed and bias states are subsequently marginalized out.

sparse for efficient solutions. In fact, visual SLAM with keyframes successfully proceeds analogously: it drops entire frames with their landmark measurements.

In the case of $\mathbf{x}_{\mathrm{T}}^{c-S}$ being a keyframe, the information loss of simply dropping all keypoint measurements would be more significant: all relative pose information between the oldest two keyframes encoded in the common landmark observations would be lost. Therefore, we additionally marginalize out the landmarks that are visible in $\mathbf{x}_{\mathrm{T}}^{k_1}$ but not in the most recent keyframe or newer frames. This means, respective landmark observations in the keyframes $k_1, \dots k_{M-1}$ are included in the linearized error term prior to landmark marginalization. Figure 5.19 depicts this procedure graphically.

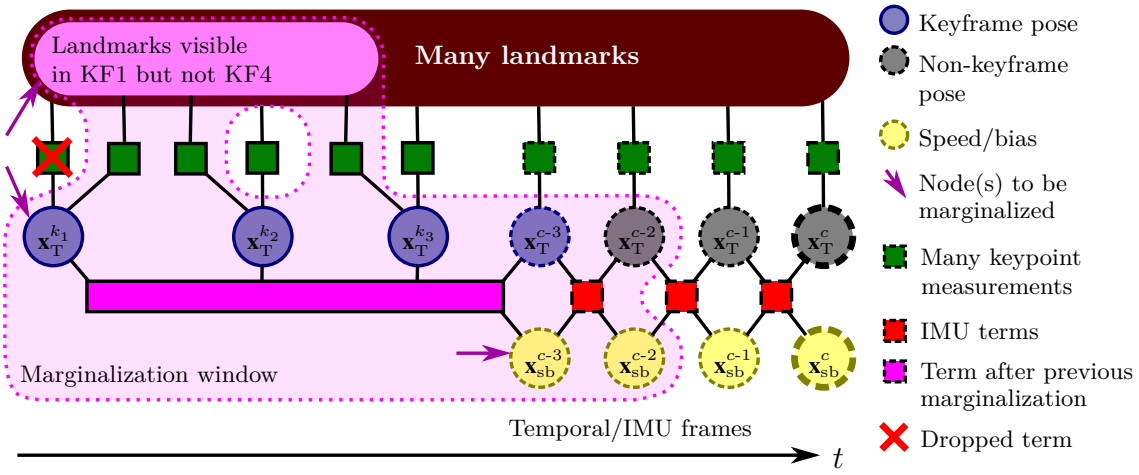


Figure 5.19: Graph for marginalization of $\mathbf{x}_{\mathrm{T}}^{c-3}$ being a keyframe: the first (oldest) keyframe ($\mathbf{x}_{\mathrm{T}}^{k_1}$) will be marginalized out. Hereby, landmarks visible exclusively in $\mathbf{x}_{\mathrm{T}}^{k_1}$ to $\mathbf{x}_{\mathrm{T}}^{k_{M-1}}$ will be marginalized as well.

The sparsity of the problem is again preserved; we show the sparsity of the Hessian matrix in Figure 5.20, along with further explanations on how measurements are dropped and landmarks may be marginalized.

The above explanations did not include extrinsics calibration nodes. The framework, owing to its generic nature, is nevertheless extended to handle this case in a straightforward manner: in fact, the extrinsics poses will also be added to the linear term, as soon as landmarks are marginalized out. In Section 5.5.6, we present results on online-estimation of temporally static extrinsics $\mathbf{x}_{\mathrm{C_i}}, i \in \{1, 2\}$; treating them as states, i.e. instances inserted at every frame, is equally possible. A temporal relative pose error has to be inserted in this case to model allowed changes of the extrinsics over time. Furthermore, marginalization of extrinsics nodes along with poses will be required in this case.

Priors and Fixation of Variables

As described previously, our framework allows for a purely relative representation of information that applies to the optimization window. This formulation constitutes a fundamental advantage over classical filtering approaches, where uncertainty is kept track of in an absolute manner, i.e. in a global frame of reference: with the absolute formulation, naturally, uncertainty will grow and increasingly incorrectly be represented through some form of linear propagation, leading to inconsistencies if not specifically addressed.

Furthermore, a filter will always need priors for all states when initializing, where they might be completely unknown and potentially bias the estimate. Our presented framework

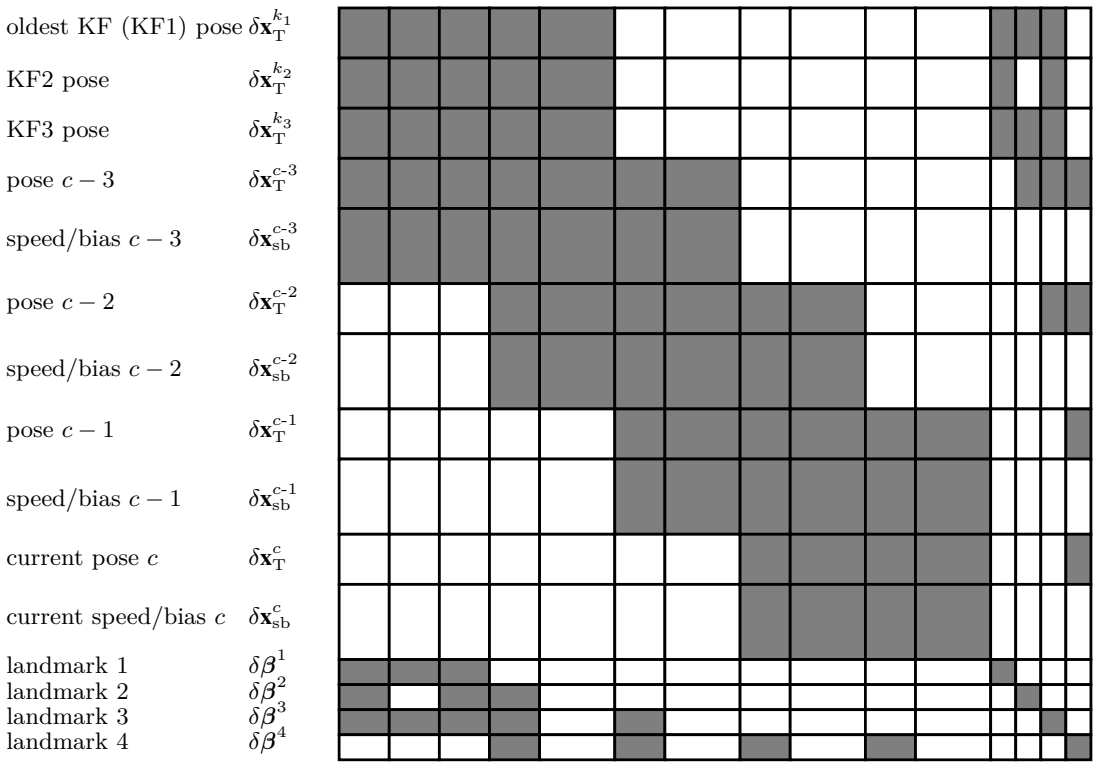


Figure 5.20: Example sparsity pattern of the Hessian matrix (gray means non-zero) for a simple case with only four landmarks. In case of marginalization of frame $c-3$, the observations of landmarks 2,3, and 4 in frame $c-3$ would be removed prior to marginalization, in order to prevent fill-in. In case of marginalization of the oldest keyframe KF1, the proposed strategy would marginalize out landmark 1, remove the observations of landmarks 2 and 3 in KF1, and leave landmark 4, since it is not observed in KF1.

does conceptually not need any priors. For more robust initialization particularly of the monocular version, however, we actually apply (rather weak) zero-mean uncorrelated priors to speed and biases. For speed we use a standard deviation of 3 m/s, which is tailored to the setups as presented in the results. For gyro bias, we applied a prior with standard deviation 0.1 rad/s and for accelerometer bias 0.2 m/s^2, which relates to the IMU parameters described in Section 5.5.6.

Inherently, the vision-only problem has 6 Degrees of Freedom (DoF) that are unobservable and need to be held fixed during optimization, i.e. the absolute pose. The combined visual-inertial problem has only 4 unobservable DoF, since gravity renders two rotational DoF observable.

In contrast to our previously published results [Leutenegger et al. 2013], we forgo fixation of absolute yaw and position: underlying optimization algorithms such as Levenberg-Marquardt will automatically cater to not taking steps along unobservable directions. Forced fixation of yaw may introduce errors, in case the orientation is not very accurately estimated.

Due to numeric noise, positive-semidefiniteness of the left-hand side linearized subpart $\mathbf{H}^*_{\lambda_1\lambda_1}$ has to be enforced at all times. To ensure this, we apply an Eigen-decomposition $\mathbf{H}^*_{\lambda_1\lambda_1} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ before optimization, and reconstruct $\mathbf{H}^*_{\lambda_1\lambda_1}$ as ${\mathbf{H}^*_{\lambda_1\lambda_1}}' = \mathbf{U}\boldsymbol{\Lambda}'\mathbf{U}^T$, where $\boldsymbol{\Lambda}'$ is obtained from $\boldsymbol{\Lambda}$ by setting all Eigenvalues below a threshold to zero.

5.5.6 Results

Throughout literature, a plethora of motion tracking algorithms has been suggested; how they perform in relation to each-other, however, is often unclear, since results are typically shown on individual datasets with differing motion characteristics as well as sensor qualities. In order to make a strong argument for our presented work, we will thus compare it to a state-of-the art visual-inertial stochastic cloning sliding-window filter which follows the MSCKF derivation of Mourikis et al. [2009].

Evaluation Setup

In the following, we provide a short overview of the hardware and settings used for dataset acquisition, as well as of the hardware and algorithms used for evaluation.

Sensor unit overview. The custom-built visual-inertial sensor is described in detail in Nikolic et al. [2014]. In essence, the assembly as shown in Figure 5.21 consists of an

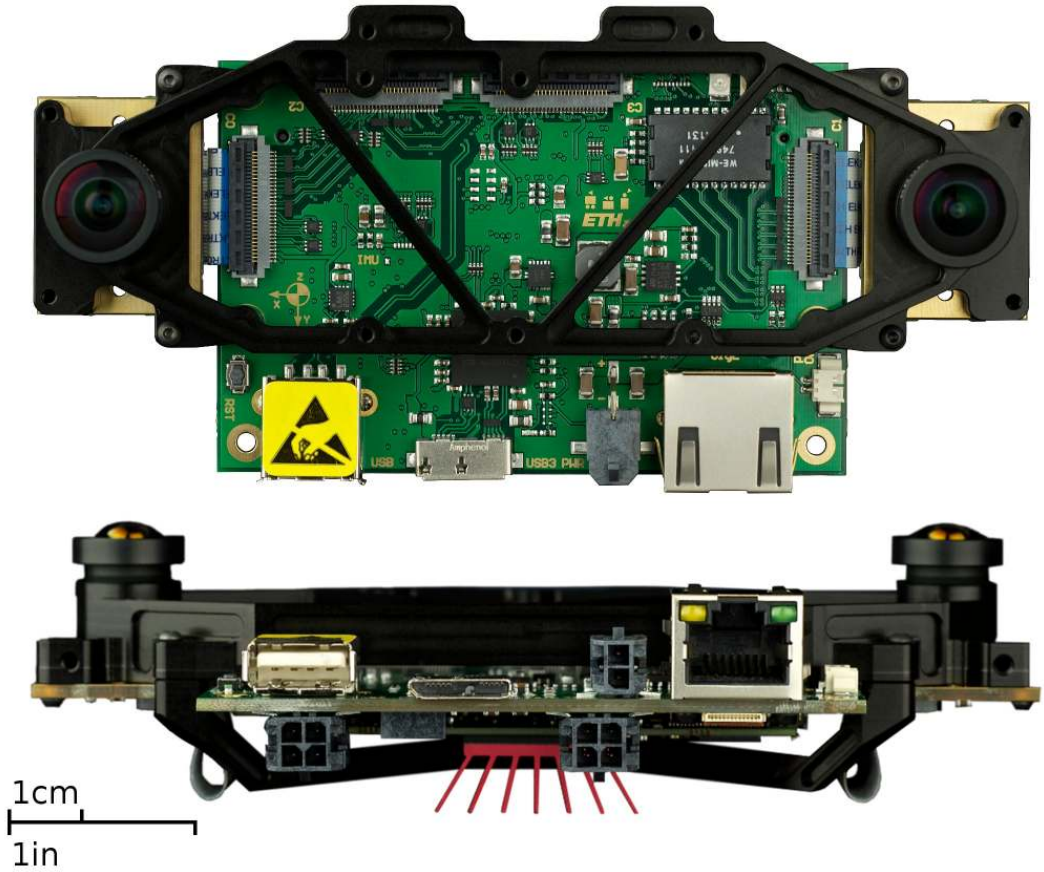


Figure 5.21: Visual-inertial sensor front and side view. Stereo imagery is hardware-synchronized with the IMU measurements and transmitted to a host computer via gigabit Ethernet.

ADIS16448 MEMS IMU and two embedded WVGA monochrome cameras with an 11 cm baseline that are all rigidly connected by an aluminum frame. An FPGA board performs hardware synchronization between imagery and IMU up to the level of pre-triggering the cameras according to the variable shutter opening times. Furthermore, the FPGA

may perform keypoint detection, in order to save CPU usage for subsequent algorithms. The data is streamed to a host computer via Gigabit Ethernet. The datasets used in this work were collected at an IMU rate of 800Hz, while the camera frame rate was set to 20 Hz (although the hardware would allow up to 60 Hz).

Sensor characteristics. We have taken the IMU noise parameters from the ADIS16448 datasheet[4] and verified them in stand-still. In order to account for unmodeled and dynamic effects, slightly more conservative numbers as listed in Table 5.2 were used. Concerning

Rate gyros			Accelerometers		
$\sigma_{c,\mathrm{g}}$	1.2e-3	$\mathrm{rad}/(\mathrm{s}\sqrt{\mathrm{Hz}})$	$\sigma_{c,\mathrm{a}}$	8.0e-3	$\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
σ_{c,b_g}	2.0e-5	$\mathrm{rad}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$	σ_{c,b_a}	5.5e-5	$\mathrm{m}/(\mathrm{s}^3\sqrt{\mathrm{Hz}})$
			τ	∞	s

Table 5.2: ADIS16448 IMU characteristics

image keypoints, we applied a detection standard deviation of 0.8 pixels. Note that the keypoints were extracted on the highest resolution image of the pyramid only. Again, this is slightly higher than what error statistics of our sub-pixel resolution Harris corner detector would suggest.

An intrinsics and extrinsics calibration (distortion coefficients and $\boldsymbol{T}_{SC_i}$) was obtained using the method described in Furgale et al. [2013].

Quantitative evaluation procedures. Defining the system boundaries of a specific algorithm along with its inputs and outputs poses some trade-offs. We chose to feed all algorithms with the same correspondences (i.e. keypoint measurements with landmark IDs per image) as they were generated by our stereo-algorithm. Each algorithm evaluated from there was left with the freedom to apply its own outlier rejection and landmark-triangulation. Obviously, all algorithms were provided with the same IMU measurements. To ensure fairness, we furthermore apply the same keypoint detection uncertainty as well as IMU noise densities, and gravity acceleration across all algorithms. Note that all parameters were left unchanged throughout all datasets and for all algorithms, including keypoint detection and matching thresholds.

We adopt the evaluation scheme of Konolige et al. [2011b] and Geiger et al. [2012]: for many starting times, the ground truth and estimated trajectories are aligned and the error is evaluated for increasing distances traveled from there.

Consider the ground-truth trajectory $\boldsymbol{T}_{GV}^{p}$ and estimated trajectory $\overline{\boldsymbol{T}}_{WS}^{p}$ both resampled at the same rate indexed by p, where $\underrightarrow{\mathcal{F}}_V$ denotes the body frame of the ground truth trajectory. Let d^p be the (scalar) distance traveled since start-up; in order to obtain a statistical characterization we choose many starting pose indices p_s such that they are spaced by a specific traveled distance. Relative to those starting poses, we define the error transformation as

$$\Delta\boldsymbol{T}(\Delta d) = \overline{\boldsymbol{T}}_{WS}^{p}\boldsymbol{T}_{SV}\boldsymbol{T}_{GV}^{p}{}^{-1}\boldsymbol{T}_{GW}^{p_s}, \forall p > p_s, \tag{5.69}$$

[4] ADIS16448 MEMS IMU datasheet available at www.analog.com/en/mems-sensors/mems-inertial-measurement-units/adis16448/products/product.html as of March 2014.

Where $\Delta d = d^p - d^{p_s}$. The many errors $\Delta \boldsymbol{T}(\Delta d)$ can now be accumulated in bins of distance traveled, in order to obtain error statistics.

At this point, we have to distinguish between the availability of 6D ground truth from the indoor Vicon motion tracking system[5] or 3D (DGPS) obtained using a Leica Viva GS14[6] ground truth.

In the 6D-case, we first have to estimate the transformation between the tracked body frame $\underrightarrow{\mathcal{F}}_V$ and the estimation body frame $\underrightarrow{\mathcal{F}}_S$. The alignment of estimator world frame $\underrightarrow{\mathcal{F}}_W$ and ground-truth world frame $\underrightarrow{\mathcal{F}}_G$ at starting time index p_s becomes trivially $\boldsymbol{T}_{GW}(p_s) = \boldsymbol{T}_{GV}(p_s)\boldsymbol{T}_{SV}^{-1}\overline{\boldsymbol{T}}_{WS}(p_s)^{-1}$.

In the 3D ground-truth case, however, we have to set $\mathbf{C}_{GV} = \mathbf{I}$. We furthermore neglect the offset between GPS antenna and IMU center (this is in the order of centimeters) and set $\boldsymbol{T}_{SV}^{-1} = \mathbf{I}$. Now the alignment of the world frames $\boldsymbol{T}_{GW}(p_s)$ is solved for as an SVD-based trajectory alignment, where the (small but large enough) segment used in this process is obviously discarded for evaluation.

Evaluation on Complementary Datasets

In the following, we will present evaluation results on three datasets. In order to cover different conditions, care was taken to record datasets with different lengths, distances to structure, speeds, dynamics, as well as with differences in illumination and number of moving objects. The main characteristics are summarized in Table 5.3. We compare both our monocular version (*aslam-mono*) and the stereo variant (*aslam*) to the (monocular) MSCKF (*msckf-mono*). Our algorithm used $M = 7$ keyframes and $S = 3$ most current frames in all datasets, while the MSCKF was set up to maintain 5 pose clones. Note

Name	Length	Duration	Max.Speed	Ground Truth
Vicon Loops	1200 m	14 min	2.0 m/s	Vicon 6D, 200 Hz
Bicycle	7940 m	23 min	13.1 m/s	DGPS 3D, 1 Hz
Around ETH	620 m	6:40 min	2.2 m/s	DGPS 3D, 1 Hz

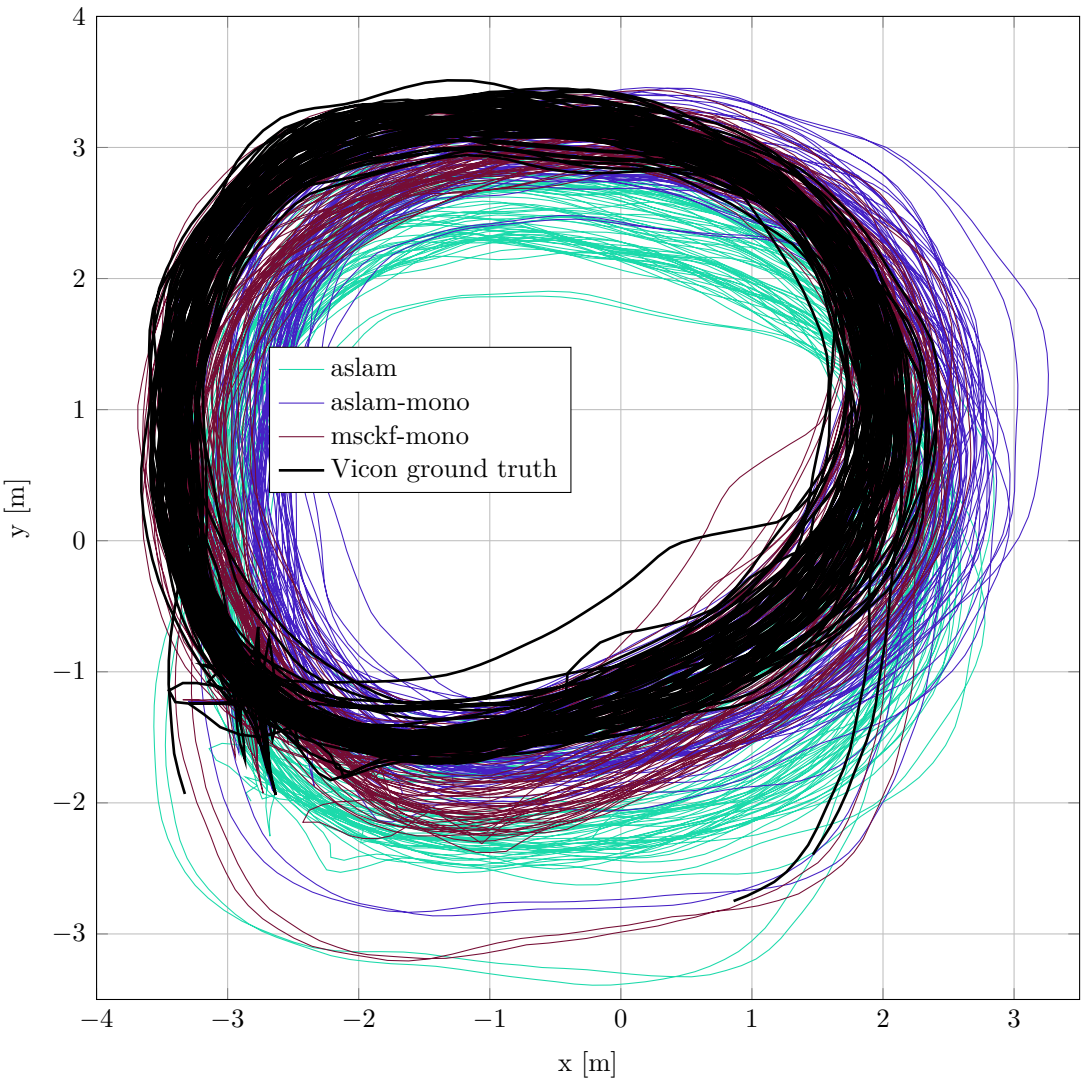
Table 5.3: Dataset characteristics.

that the following trajectory reconstructions do not include sigma-bounds; this is related to the fact that the presented framework only uses relative information and thus global uncertainty is not represented. While we consider this formulation a major advantage of our approach, it comes at the cost of not being able to plot uncertainty bounds. We regard global uncertainty representation in an incremental, keyframe-to-keyframe manner as part of future work.

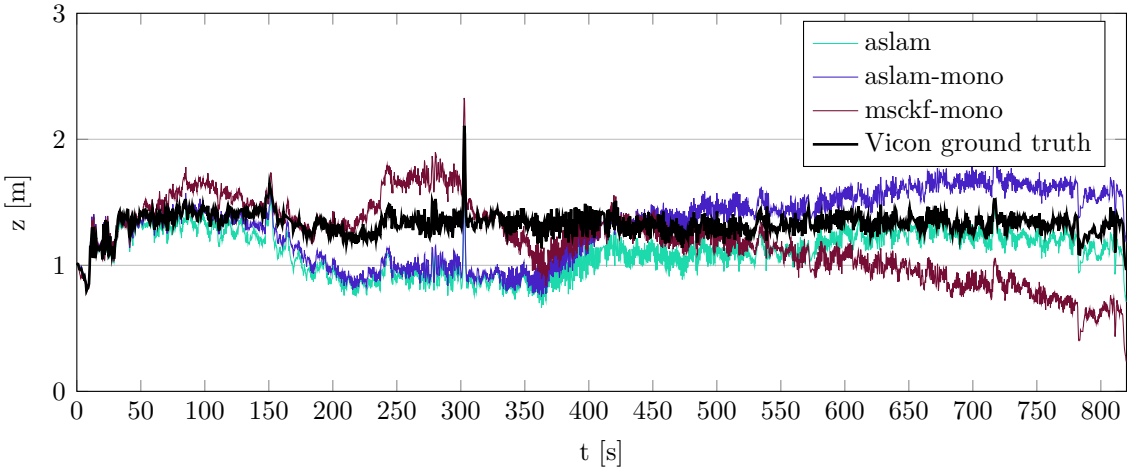
Vicon loops. A trajectory was recorded with the handheld sensor inside our Vicon room. Consequently, the path is very much limited in spatial extent, while only close structure is observed. Full 6D ground-truth is available from external motion tracking at 200 Hz. The sequence lasts almost 14 minutes, walking mostly in circles no faster than 2.0 m/s. We show the overhead plot and altitude profile in Figure 5.22 along with absolute errors as a function of distance traveled in Figure 5.23.

[5] Vicon motion tracking system, see www.vicon.com as of March 2014.

[6] Leica Viva GS14 GNSS recorder www.leica-geosystems.com/en/Leica-Viva-GS14_102200.htm as of March 2014.



(a) Overhead plot of the vicon dataset.



(b) Altitude profiles, i.e. z-components of estimator outputs represented in $\underrightarrow{\mathcal{F}}_G$.

Figure 5.22: Vicon dataset evaluation: lateral and vertical positions.

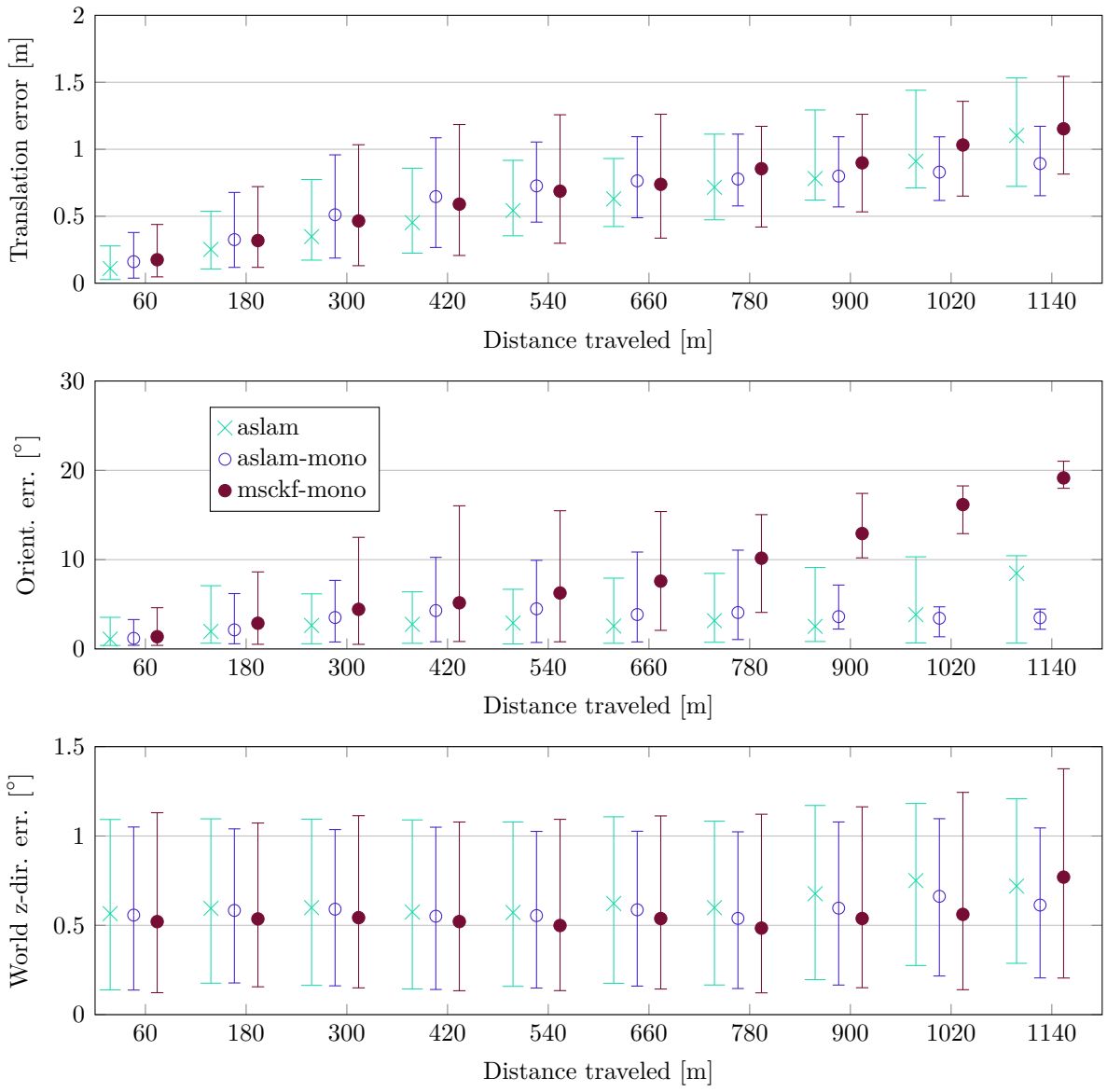
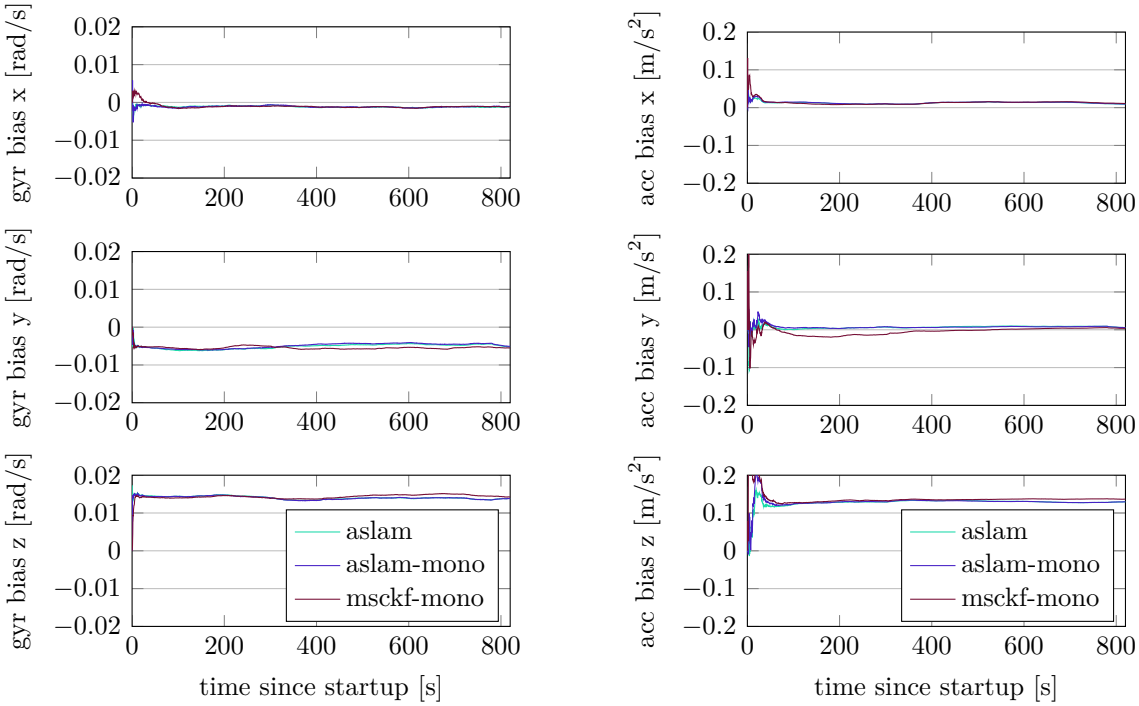


Figure 5.23: Error statistics in terms of median, 5^{th}, and 95^{th} percentiles: norm of position error (top), norm of error axis angle vector (middle), and angle between ground truth down axis and estimator down axes (bottom). While differences in position errors between the different algorithms and variants are not very significant, yaw drift of the *msckf* is clearly higher.

Note that all algorithms achieve below 0.1% of median position error per distance traveled at the end of the 1200 m long path; this is, however, not only caused by the high accuracy of the algorithms. In fact, yaw drift, which is clearly present, does not become manifest as much in position error as in the other datasets that cover larger distances. The algorithms do not show extreme differences, but some subtleties may nonetheless be identified: while all manage to estimate the World z-direction (aligned with acceleration due to gravity), the computationally more expensive algorithms proposed in this work expectedly show a slightly better performance in terms of yaw drift. We furthermore provide the plots of both gyro and accelerometer biases in Figure 5.24. Note that despite their different natures, all algorithms converge to tracking very similar values.



(a) Gyro bias estimates.

(b) Accelerometer bias estimates.

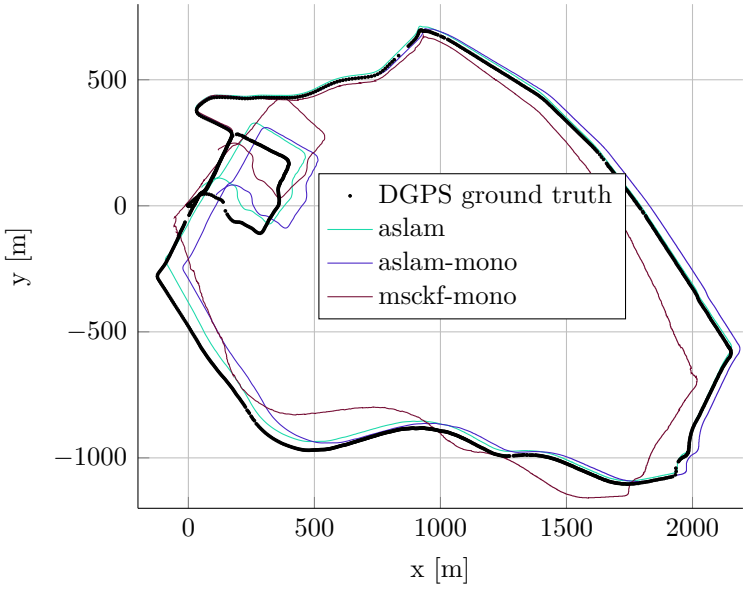
Figure 5.24: Evolution of bias estimates by the different algorithms: despite the different characteristics of the algorithms, they converge to and track similar values.

Bicycle trajectory. The sensor was mounted onto a helmet and worn for a bicycle ride of 7.9 kilometers from ETH Hönggerberg into the city of Zurich and back to the starting point. Figure 5.25 illustrates the setup. Speeds up to 13 m/s were reached during

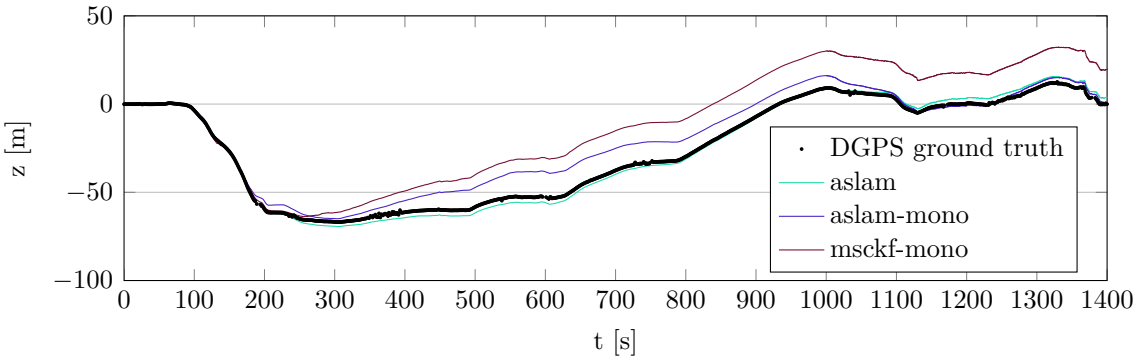


Figure 5.25: Simon Lynen ready for dataset collection on a bicycle at ETH with sensor and GNSS ground-truth recorder mounted to a helmet.

the 23 minute long course. Post-processed DGPS ground truth is available at 1 Hz, and all measurements with a position uncertainty beyond 1 m were discarded. Figure 5.26 displays reconstructed trajectories as compared to ground-truth and Figure 5.27 reports the statistics on the position error normalized by distance traveled.



(a) Overhead plot of the reconstructed bicycle ride trajectories.



(b) Altitude profile of bicycle ride.

Figure 5.26: Trajectories of the Bicycle Trajectory dataset: the filtering approach *msckf-mono* accumulates the largest yaw error that becomes manifest also in position error. As expected, our stereo variant performs the best, which is also apparent in the altitude evolution.

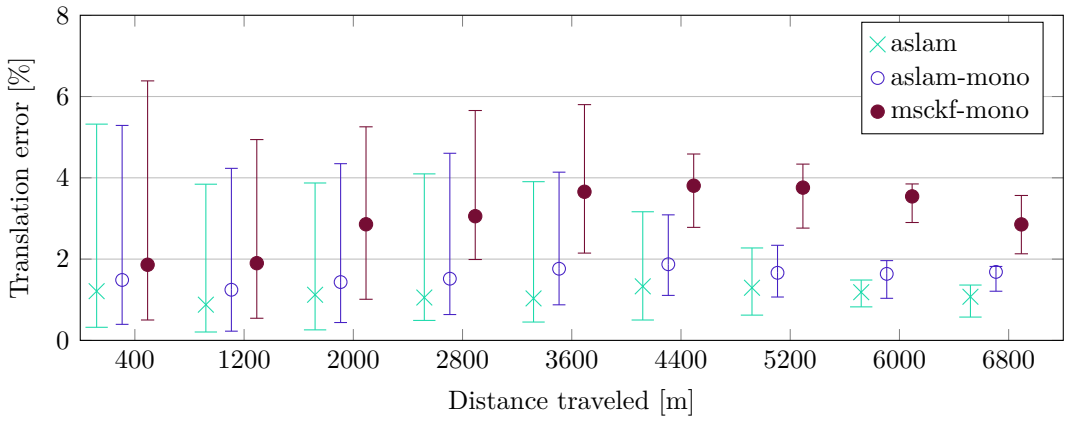
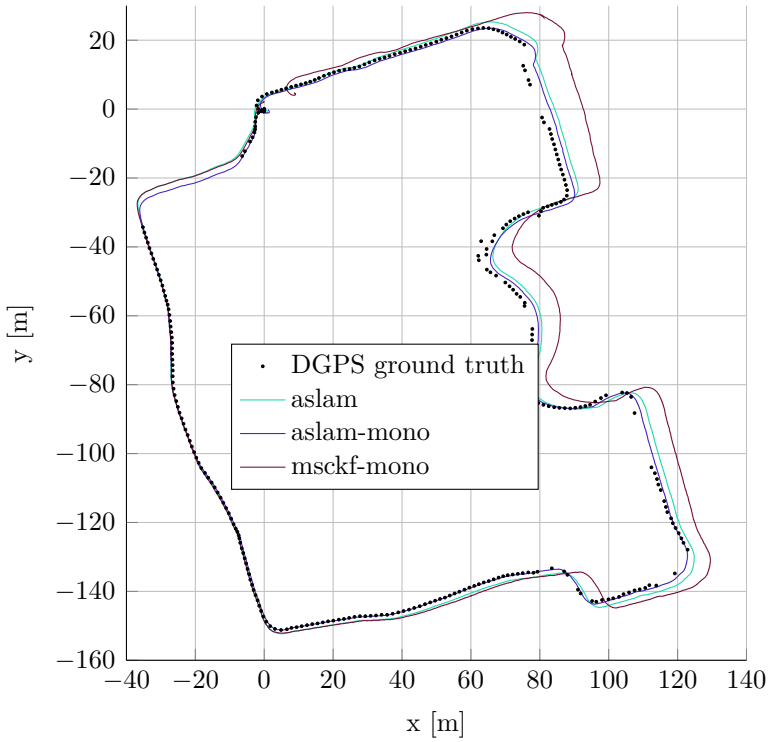


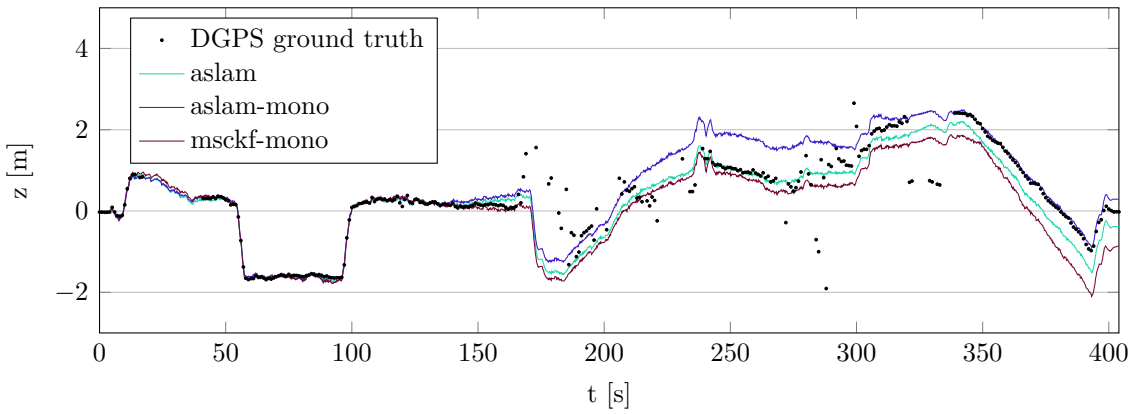
Figure 5.27: Relative position error statistics: median, 5$^{\text{th}}$, and 95$^{\text{th}}$ percentiles.

As expected, the stereo-version shows a notably better performance than the monocular one. Both outperform the MSCKF, which clearly suffers from more yaw drift. It is furthermore worth mentioning that *aslam-mono* and *aslam* accumulate less drift in altitude, where a clear advantage of the stereo algorithm becomes visible.

Handheld around ETH Main Building. This second outdoor loop was recorded with the handheld sensor while walking around the main building of central ETH (no faster than 2.2 m/s). The path length amounts to 620 m. The imagery is characterized by varying depth of the observed structure, and includes some pedestrians walking by. Figures 5.28 and 5.29 summarize the results. Again, both our approaches outperform the



(a) Overhead plot of the reconstructed trajectories.



(b) Altitude profile.

Figure 5.28: Evaluation results for the ETH Main Building dataset: interestingly, the stereo version of our algorithm is outperformed by the mono variant. The cause is further investigated in Section 5.5.6.

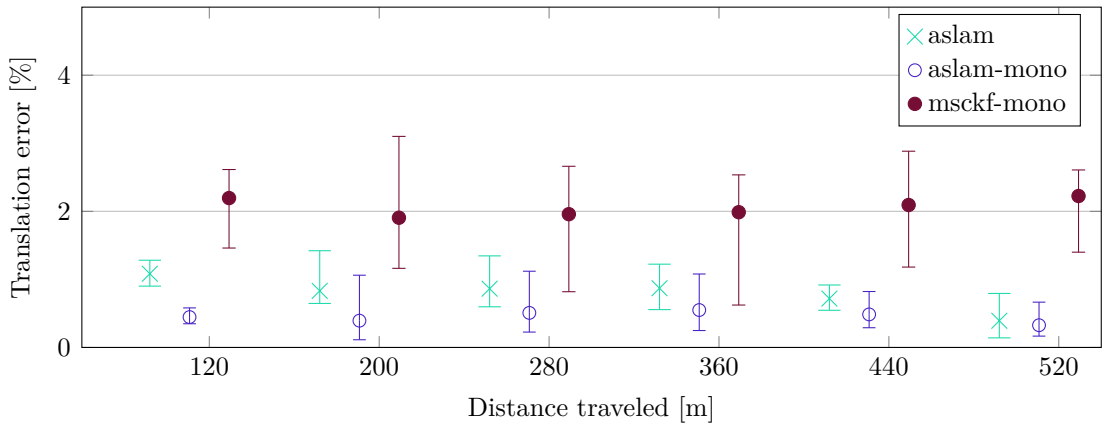


Figure 5.29: Relative position error statistics: median, 5^{th}, and 95^{th} percentiles.

reference implementation of the MSCKF. Qualitatively, the yaw error seems to contribute the least to the position error; rather the position drift appears to originate from (locally) badly estimated scale. Interestingly, the stereo version of our algorithm seems to perform slightly worse in this respect. We suspect small errors in the stereo calibration to cause this behavior, concretely a slight mismatch of relative camera orientation. This issue is further investigated in Section 5.5.6.

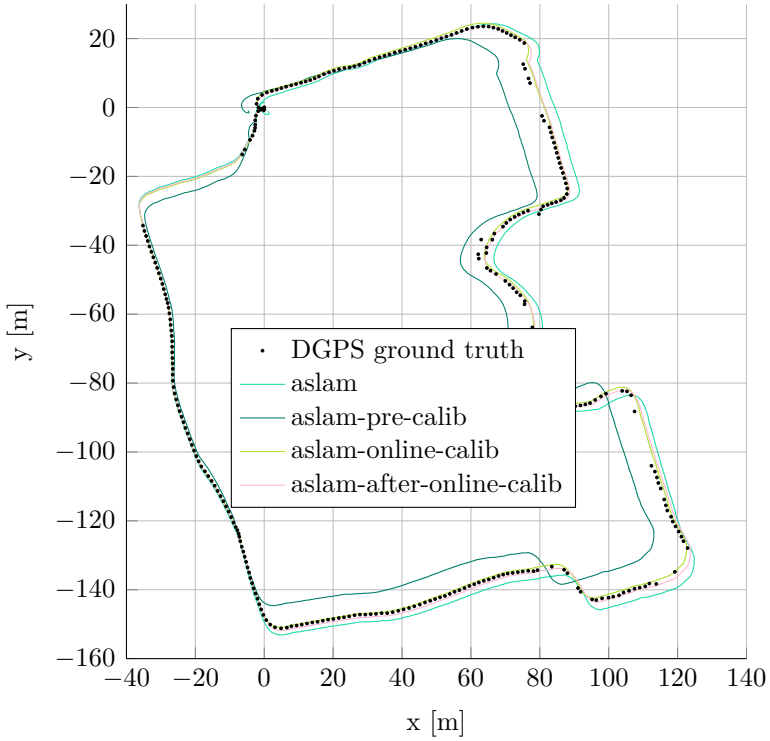
Parameter Studies

With focus on the proposed algorithm, studies are provided that investigate the sensitivity on the performance w.r.t. a selected set of parameters.

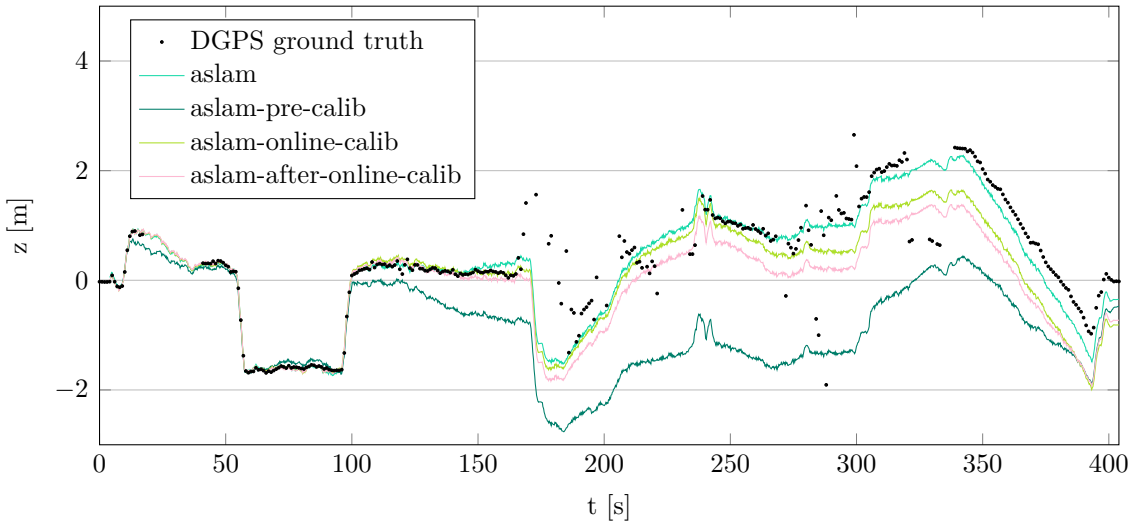
Online extrinsics calibration. In the following experiment, we assess the performance of our online IMU to camera extrinsics estimation scheme. We assume a starting point of calibrated intrinsics, for which off-the-shelf tools exist; and we furthermore take rough extrinsics as available in a straightforward manner e.g. from a Computer Aided Design (CAD) software. For the problem to be best constrained, we run the stereo algorithm.

In order to obtain a well-defined optimization problem, we apply a weak prior to all extrinsic translations (10 mm standard deviation) as well as orientation (0.6° standard deviation).

Using the ETH Main Building dataset as introduced above, Figures 5.30 and 5.31 display a respective comparison in terms of estimation accuracy of pre-calibration, online-calibration and post-calibration to the result as shown above with the original calibration. Remarkably, the very rough extrinsics guess generates mostly scale mismatch, while the orientation seems consistent. In fact, the scale is not simply wrong due to incorrect baseline setting—since the baseline is known and set to a much higher accuracy than the 10% scale error. Interestingly, the estimates during online calibration are significantly more accurate than with fixed original calibration. [Konolige et al. 2011b] found that their online extrinsics estimation with loose coupling of IMU measurements was biased to the type of scenery and camera placement. Thus we suspect that the checkerboard calibration in the office might not represent the the optimal setup for best performance in typical scenes later on (i.e. larger distance to structure). In fact the slight scale mismatch is completely removed. Moreover, taking the frozen final online estimates



(a) Overhead plot with extrinsics calibration.



(b) Altitude profile with extrinsics calibration.

Figure 5.30: Evaluation results for the extrinsics calibration. Compared to the original estimate (*aslam*), a reconstruction with fixed roughly aligned extrinsics (*aslam-pre-calib*) yields expectedly rather poor results. Estimates during online calibration (*aslam-online-calib*) that use the rough alignment as starting guess manages to even outperform the original result. Freezing the online estimates to the final values and re-running the process (*aslam-after-online-calib*) results in equivalent performance as with online calibration turned on.

and re-running the process results in no significant change as compared to the online estimation—suggesting that online extrinsics calibration may be safely left switched on, at least in the stereo case. In such a case, however, noise would need to be injected into the estimation process, in order to allow for the extrinsics to (slowly) drift. Figure 5.32 addresses the remaining question of whether or not the extrinsics converge, and if the

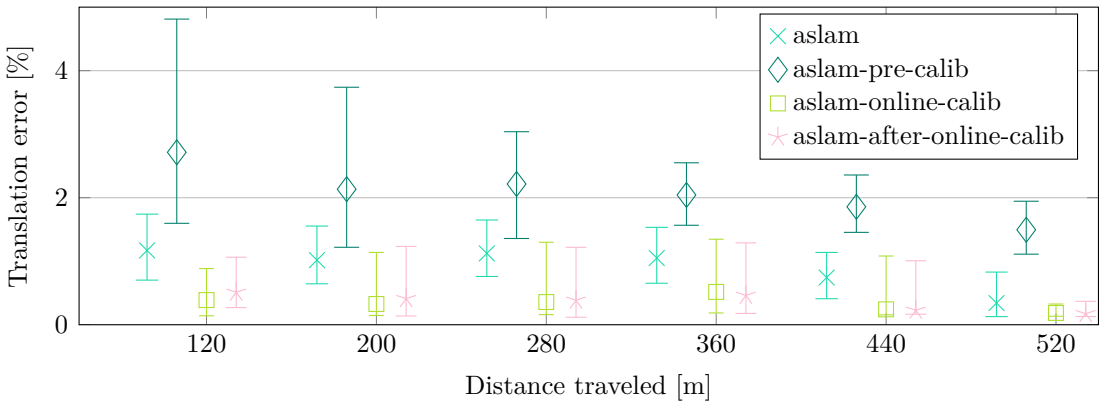


Figure 5.31: Relative error statistics of the extrinsics calibration: median, 5^{th}, and 95^{th} percentiles. The qualitative impression from the trajectory plots is confirmed.

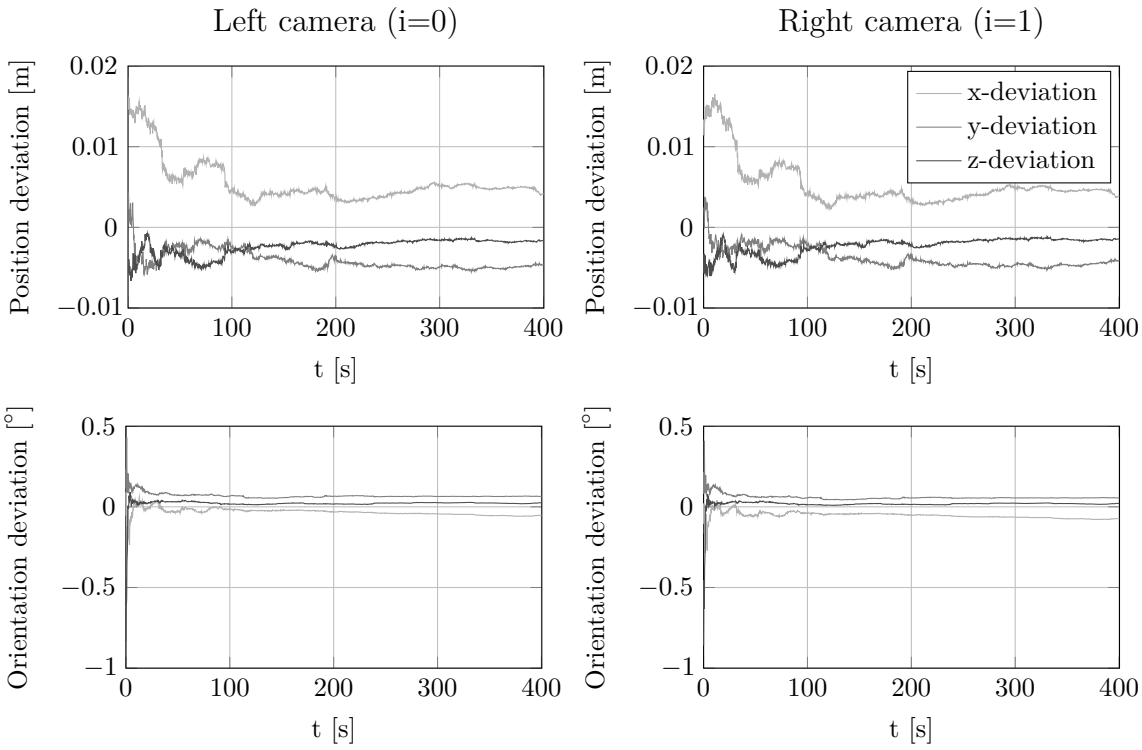


Figure 5.32: Position differences of online calibrated extrinsics translation (top) and axis-angle orientation differences w.r.t. original (bottom) for left and right cameras.

respective estimates correspond to the original calibration. Clearly, the estimates of the IMU to camera orientations converge fast to a stable estimate, while the camera positions with respect to the IMU that are less well observable are subjected to more mobility.

Influence of keyframe and keypoint numbers. We claim that the proposed algorithm offers scalability in terms of tailoring it according to the trade-off between accuracy and processing constraints. In this context, we analyze the influence of two main parameters on the performance: on the one hand, we can play with the number of pose variables to be estimated, and on the other hand we have the choice of average number

of landmark observations per image by adjusting the keypoint detection threshold.

Taking the ETH Main Building dataset and running the mono-version of our algorithm, we show the quantitative results for different keyframe number M settings in Figure 5.33. In the same comparison, we furthermore varied the number of frames connected by nonlinear IMU terms S and provide the full-batch solution as a gold standard reference. Note that the full-batch problem was initialized with the original version or our algorithm and run to convergence. At the low end of the frame number ($M = 4$), we see a clear

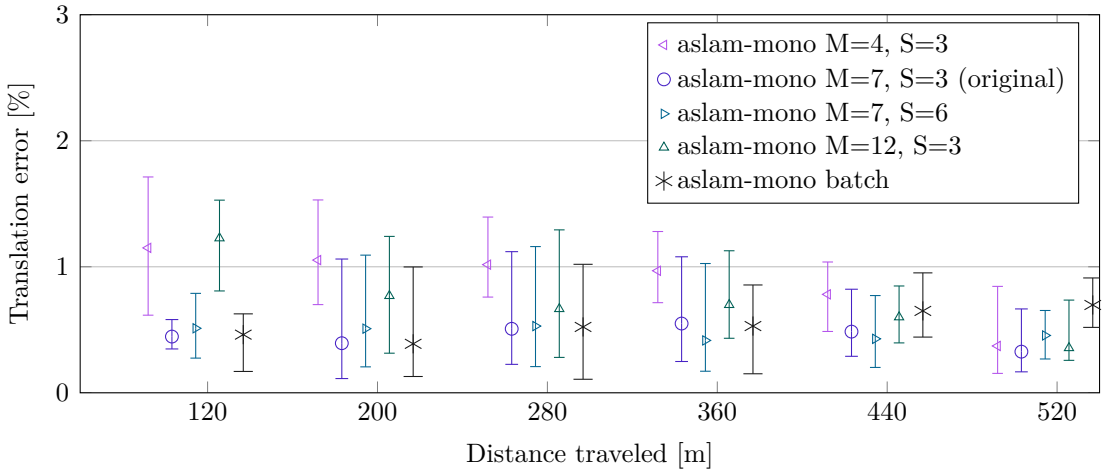


Figure 5.33: Comparison of different frame number (N keyframes and S IMU frames) settings with respect to the full batch solution (initialized with the result from $N = 7, S = 3$).

performance drawback, whereas large numbers of keyframes $M = 12$ do not seem to increase accuracy. Another interesting finding is that increasing S to account for more nonlinear error terms does not become manifest in less error. Also note that the full-batch optimization does not increase the overall accuracy of the solution—indicating that the approximations in terms of linearization, marginalization, as well as measurement dropping as described here are forming a suite of reasonable choices.

Note that in the overall complexity of the algorithm, the number of keyframes M contributes with $\mathcal{O}(M^3)$ when it comes to solving the respective dense part of the linear system of equations.

Finally, we also investigate the influence of the keypoint detection threshold u that directly affects keypoint density in image space. Figure 5.34 summarizes the respective quantitative results, again processing the ETH Main Building dataset with the monocular version of our algorithm. Interestingly, all versions perform similarly on shorter ranges, despite the large variety in average keypoints per image, i.e. 45.3, 110.3, and 239.2. A slight trend suggesting that more keypoints result in better pose estimates is only visible for longer traveled distances. Note that increasing the detection threshold inevitably not only decreases execution time, but also comes at the expense of environment representation richness in terms of landmark density.

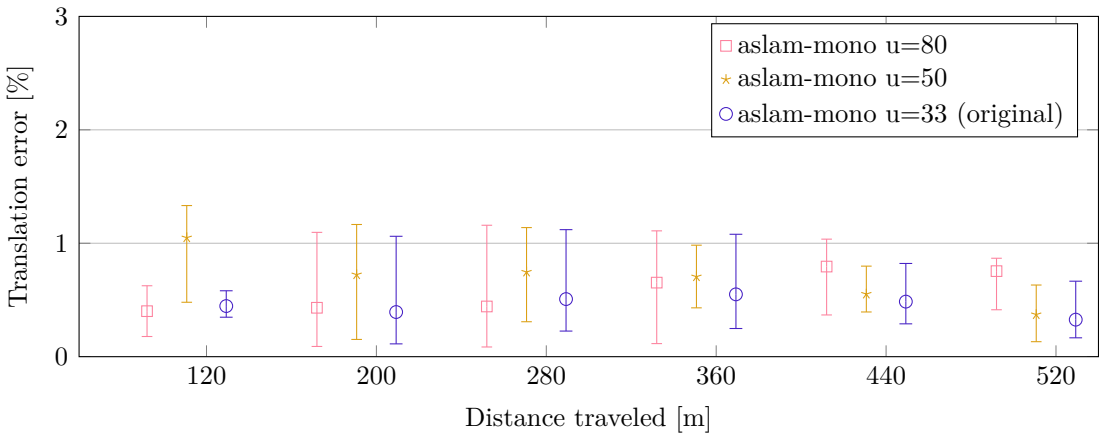


Figure 5.34: Comparison of different keypoint detection thresholds $u \in \{80, 50, 33\}$ settings. The corresponding mean number of keypoints per image are 45.3, 110.3, and 239.2 in this dataset.

5.6 Discussion

We have introduced models for various sensors and methods describing how they can be fused in a rigorously probabilistic manner, both in filtering as well as in nonlinear optimization. Aiming at long-term UAS deployment, two estimators have been presented in detail, which have indirect IMU propagation in common.

A first fusion strategy formulated as an EKF uses updates from GPS, a three-axis magnetometer, as well as dynamic and static pressure measurements. Owing to its light-weight nature, it may be implemented on a simple micro-controller. It was shown how it can deal with outliers and GPS outages. The filter forms the basis for ASL's autonomous unmanned airplane waypoint navigation.

Using visual cues in the estimation process offers two key benefits: first of all, it allows for awareness of the surrounding, which may constitute an imperative necessity for certain UAS deployment scenarios. Second, very accurate relative spatial correlations can be exploited, complementing the strong time correlations provided by the IMU. We suggested a respective fusion framework that employs the concepts of nonlinear optimization and keyframing resulting in high-accuracy outputs that may still be run in real-time on today's embedded computers.

5.6.1 Multi-Sensor Fusion without Computer Vision

A robust state estimation framework for airplanes was presented that is based on Extended Kalman Filtering. The generality of the proposed framework makes it applicable to both unmanned and manned airplanes. Not only inertial sensors, magnetometers and GPS updates are used, but also both static and dynamic pressure measurements. In the state, we estimate position and orientation, velocity and IMU biases, as well as the 3D wind vector and the pressure at sea level.

The resulting filter is robust in the sense that it can cope with even long GPS outage. Such failures might occur only rarely with today's GPS receivers, but a resulting lack of attitude and heading information can be catastrophic. We regard temporary roll angles larger than 90°, in-flight reset (watchdog reset, power loss) as well as long lasting unavailability as realistic outage scenarios. Under outage conditions, orientation

is continued to be estimated accurately and as soon as GPS is recovered, a seamless transition back to normal operation is made. We demonstrated the real-world applicability of our approach on real flight data obtained with low-cost electronics as well as by the successful on-board implementation.

Future work will concern the inclusion of the full sensor calibration and further evaluation.

5.6.2 Using Camera Measurements

We have introduced a framework of tightly coupled fusion of inertial measurements and image keypoints in a nonlinear optimization problem that applies linearization and marginalization in order to achieve keyframing. As an output, we obtain poses, velocities, and IMU biases as a time series, as well as a local 3D map of sparse landmarks. The proposed algorithm is bounded in complexity, as the optimization includes a fixed number of poses.

The keyframing, in contrast to a fixed-lag smoother, allows for arbitrary temporal extent of estimated camera poses jointly observed landmarks. As a result, our framework achieves high accuracy, while still being able to operate at real-time. We showed extensive evaluation of both a stereo and a mono version of the proposed algorithm on complementary datasets with varied type of motion, lighting conditions, and distance to structure. In this respect, we made the effort to compare our results to the output of a state-of-the-art visual-inertial stochastic cloning sliding-window filter, which follows the MSCKF algorithm and is fed with the same IMU data and keypoints with landmark associations. While admittedly being computationally more demanding, our approach consistently outperforms the filter.

In further studies we showed how online-calibration of camera extrinsics can be incorporated into our framework: results on the stereo version indicate how slight miscalibration can become manifest in scale error; online calibration, even starting from a very rough initial guess, removes this effect. Finally, we also address scalability of the proposed method in the sense of tailoring to hardware characteristics, and how the setting of number of frames as well as detected keypoints affect accuracy. Interestingly, employing larger numbers of keyframes, such as 12, does not show a significant advantage over the standard setting of 7, at least in exploratory motion mode. Furthermore, we don't observe a dramatic performance decrease when reducing average numbers of keypoints per image from 240 to 45.

The way is paved for deployment of our algorithm on various robotic platforms such as Unmanned Aerial Systems. In this respect, we are planning to release our proposed framework as an open-source software package. Moreover, we will explore inclusion of other, platform-specific sensor feeds, such as wheel odometry, GPS, magnetometer or pressure measurements—with the aim of increasing accuracy and robustness of the estimation process, both primary requirements for successful deployment of robots in challenging real environments.

Chapter 6
BRISK: Efficient Image Keypoints

Effective and efficient generation of keypoints from an image is a well-studied problem in the literature and forms the basis of numerous Computer Vision applications, including data association for vision-supported odometry or Simultaneous Localization and Mapping. Established leaders in the field are the SIFT and SURF algorithms which exhibit great performance under a variety of image transformations, with SURF in particular considered as the most computationally efficient amongst the high-performance methods to date.

In this chapter, we propose BRISK: Binary Robust Invariant Scalable Keypoints, a novel method for keypoint detection, description and matching. A comprehensive evaluation on benchmark datasets reveals BRISK's adaptive, high descriptive power as in state-of-the-art algorithms, albeit at a dramatically lower computational cost (an order of magnitude faster than SURF in cases). The key to speed lies in the application of a novel scale-space corner detector in combination with the assembly of a bit-string descriptor from intensity comparisons retrieved by dedicated sampling of each keypoint neighborhood.

6.1 Introduction

Decomposing an image into local regions of interest or *'features'* is a widely applied technique in Computer Vision used to alleviate complexity while exploiting local appearance properties. Image representation, object recognition and matching, 3D scene reconstruction and motion tracking all rely on the presence of stable, representative features in the image, driving research and yielding a plethora of approaches to this problem.

The ideal keypoint detector finds salient image regions such that they are *repeatably* detected despite change of viewpoint; more generally it is robust to many possible image transformations. Similarly, the ideal keypoint descriptor captures the most important and distinctive information content comprised in the detected salient regions, such that

Parts of this chapter appear in:

- S. Leutenegger and M. Chli and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555). IEEE, 2011.

the same structure can be *recognized* if encountered. Moreover, on top of fulfilling these properties to achieve the desired quality of keypoints, the speed of detection, description and matching needs also to be optimized to satisfy the time-constraints of the task at hand.

In principle, state-of-the-art algorithms target applications with either strict requirements in precision or speed of computation. Lowe's SIFT approach [Lowe 2004] is widely accepted as one of highest quality options currently available, promising distinctiveness and invariance to a variety of common image transformations—however, at the expense of computational cost. On the other end of the spectrum, a combination of the FAST [Rosten and Drummond 2006] keypoint detector and the BRIEF [Calonder et al. 2010] approach to description offers a much more suitable alternative for real-time applications. However, despite the clear advantage in speed, this latter approach suffers in terms of reliability and robustness as it has minimal tolerance to image distortions and transformations, in particular to in-plane rotation and scale change. As a result, real-time applications like SLAM [Davison et al. 2007] need to employ probabilistic methods [Chli and Davison 2008] for data association to discover matching consensus.

Figure 6.1 shows an exemplary output of our proposed algorithm in terms of keypoint-associations in an image pair exhibiting various transformations.

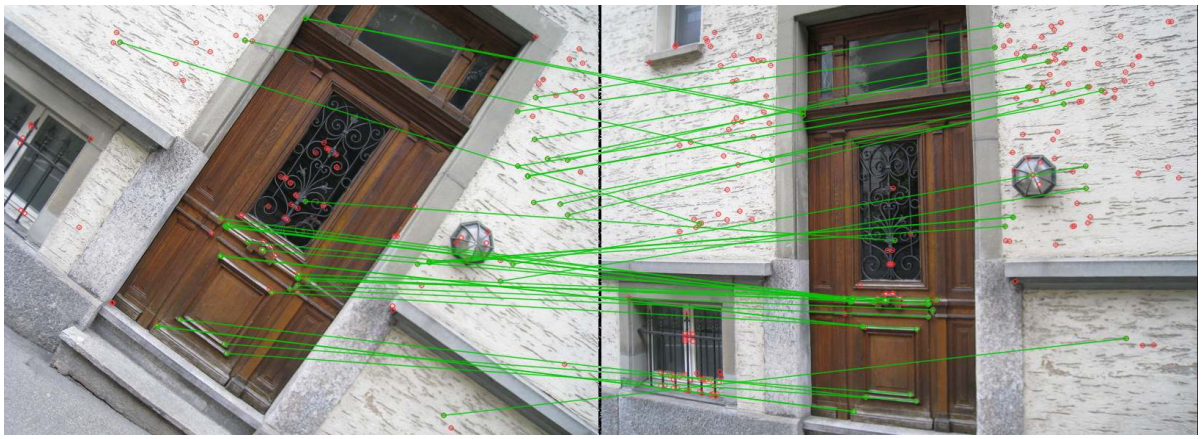


Figure 6.1: BRISK matching example: a detection threshold of 70 is used and a matching Hamming distance threshold of 90. The resulting matches are connected by the green lines and show no clear false positives.

6.1.1 Related Work

Identifying local interest points to be used for image matching can be traced a long way back in the literature, with Harris and Stephens [1988] proposing one of the earliest and probably most well-known corner detectors. The seminal work of Mikolajczyk et al. [2005] presented a comprehensive evaluation of the most competent detection methods at the time, which revealed no single all-purpose detector but rather the complementary properties of the different approaches depending on the context of the application. The more recent FAST criterion [Rosten and Drummond 2006] for keypoint detection has become increasingly popular in state-of-the-art methods with hard real-time constraints, with AGAST [Mair et al. 2010] extending this work for improved performance.

Amongst the best quality features currently in the literature is the SIFT [Lowe 2004]: the high descriptive power and robustness to illumination and viewpoint changes has

rated the SIFT descriptor at the top of the ranking list in the survey in [Mikolajczyk and Schmid 2003]. However, the high dimensionality of this descriptor makes SIFT prohibitively slow for many large-scale or real-time oriented applications. PCA-SIFT [Ke and Sukthankar 2004] reduced the descriptor from 128 to 36 dimensions, compromising however its distinctiveness and increasing the time for descriptor formation which almost annihilates the increased speed of matching. The GLOH descriptor [Mikolajczyk and Schmid 2005] is also worth noting here, as it belongs to the family of SIFT-like methods and has been shown to be more distinctive but also more expensive to compute than SIFT.

The growing demand for high-quality, high-speed features has led to more research towards algorithms able to process richer data at higher rates. Notable is the work of Agrawal et al. [2008] who apply a center-symmetric local binary pattern as an alternative to SIFT's orientation histograms approach. The most recent BRIEF [Calonder et al. 2010] is designed for super-fast description and matching and consists of a binary string containing the results of simple image intensity comparisons at random pre-determined pixel locations. Despite the simplicity and efficiency of this approach, the method is very sensitive to image rotation and scale changes, restricting its application to general tasks.

Probably the most appealing features at the moment are SURF [Bay et al. 2008], which have been demonstrated to be significantly faster than SIFT. SURF detection uses the determinant of the Hessian matrix (blob detector), while the description is formed by summing Haar wavelet responses at the region of interest. While demonstrating impressive timings with respect to the state-of-the-art, SURF are, in terms of speed, still orders of magnitude away from the fastest, yet limited versatility features currently available.

6.1.2 Outline and Contributions of BRISK

In this paper, we present a novel methodology dubbed 'BRISK' for fast keypoint detection, description and matching. As suggested by the name, the method is rotation as well as scale invariant to a significant extent, achieving performance comparable to the state of the art while dramatically reducing computational cost. Following a description of the approach, we present experimental results performed on the benchmark datasets and using the standardized evaluation method of [Mikolajczyk and Schmid 2005; Mikolajczyk et al. 2005]. Namely, we present evaluation of BRISK with respect to SURF and SIFT which are widely accepted as a standard of comparison under common image transformations.

The inherent difficulty in extracting suitable keypoints from an image lies in balancing two competing goals: high distinctiveness in description and low computational requirements. This is where this work aims to set a new milestone with the BRISK methodology. Perhaps the most relevant work tackling this problem is SURF [Bay et al. 2008] which has been demonstrated to achieve robustness and speed, only, as evident in our results, BRISK achieves comparable recall and precision in matching at much less computation time. In a nutshell, this chapter proposes a novel method for generating keypoints from an image, composed of the following steps:

- **Scale-space keypoint detection:** Points of interest are identified across both the image and scale dimensions using a saliency criterion. In order to boost efficiency of computation, keypoints are detected in octave layers of the image pyramid as well

as in layers in-between. The location and the scale of each keypoint are obtained in the continuous domain via quadratic function fitting.

- **Keypoint description:** A sampling pattern consisting of points lying on appropriately scaled concentric circles is applied at the neighborhood of each keypoint to retrieve grayscale values: processing local intensity gradients, the feature characteristic direction is determined. Finally, the oriented BRISK sampling pattern is used to obtain pairwise brightness comparison results which are assembled into the binary BRISK descriptor.

- **Descriptor matching:** Once generated, the BRISK keypoints can be matched very efficiently thanks to the binary nature of the descriptor. With a strong focus on efficiency of computation, BRISK also exploits the speed savings offered in the SSE instruction set widely supported on today's architectures.

The BRISK reference implementation by the authors was released under BSD license and has now been integrated into the OpenCV library[2], available under BSD as well.

The remainder of this chapter is structured as follows: we describe the detection step in Section 6.2, followed by descriptor extraction in Section 6.3. Extensive evaluation results and parameter studies are presented in Section 6.4, including timings. Section 6.5 finally discusses the main findings and outlines possible future directions for continuing this work.

6.2 Scale-Space Keypoint Detection

Here, we describe the feature detection part of BRISK. It is, however, important to note that the modularity of the method allows the use of the BRISK detector in combination with any other keypoint descriptor and vice versa.

6.2.1 FAST and HARRIS-Based Corner Score

With the focus on efficiency of computation, our detection methodology is originally inspired by the work of Mair et al. [2010] for detecting regions of interest in the image. Their AGAST is essentially an extension for accelerated performance of the now popular FAST, proven to be a very efficient basis for feature extraction. FAST makes use of lazy evaluation heavily, i.e. it does only compute corner scores as far as needed for corners that surpass a user-provided threshold. It is important to note here that FAST and AGAST provide different alternatives of mask shapes for keypoint detection. In BRISK, we mostly use the 9-16 mask, which essentially requires at least 9 consecutive pixels in the 16-pixel circle to either be sufficiently brighter or darker than the central pixel for the FAST criterion to be fulfilled.

As an alternative basis for detection, we therefore also employ the Harris corner detector [Harris and Stephens 1988], which computes the cornerness image in a brute-force manner, allowing for easier integration with uniformity enforcement as discussed below. Moreover, the scheme results in detection times that are largely independent of the number of detected corners, which is a desirable property for real-time applications employing BRISK.

[2] See opencv.org, accessed April 2014.

6.2.2 Scale Detection

With the aim of achieving invariance to scale, which is crucial for certain applications, we go a step further by searching for maxima not only in the image plane, but also in scale-space using the corner score s as a measure for saliency. Despite discretizing the scale axis at coarser intervals than in alternative high-performance detectors (e.g. the Fast-Hessian [Bay et al. 2008]), the BRISK detector estimates the true scale of each keypoint in the continuous scale-space.

In the BRISK framework, the scale-space pyramid layers consist of n octaves c_i and n intra-octaves d_i, for $i = \{0, 1, \ldots, n-1\}$ and typically $n = 4$. The octaves are formed by progressively half-sampling the original image (corresponding to c_0). Each intra-octave d_i is located in-between layers c_i and c_{i+1} (as illustrated in Figure 6.2). The first intra-octave d_0 is obtained by downsampling the original image c_0 by a factor of 1.5, while the rest of the intra-octave layers are derived by successive halfsampling. Therefore, if τ denotes scale then $\tau(c_i) = 2^i$ and $\tau(d_i) = 2^i \cdot 1.5$.

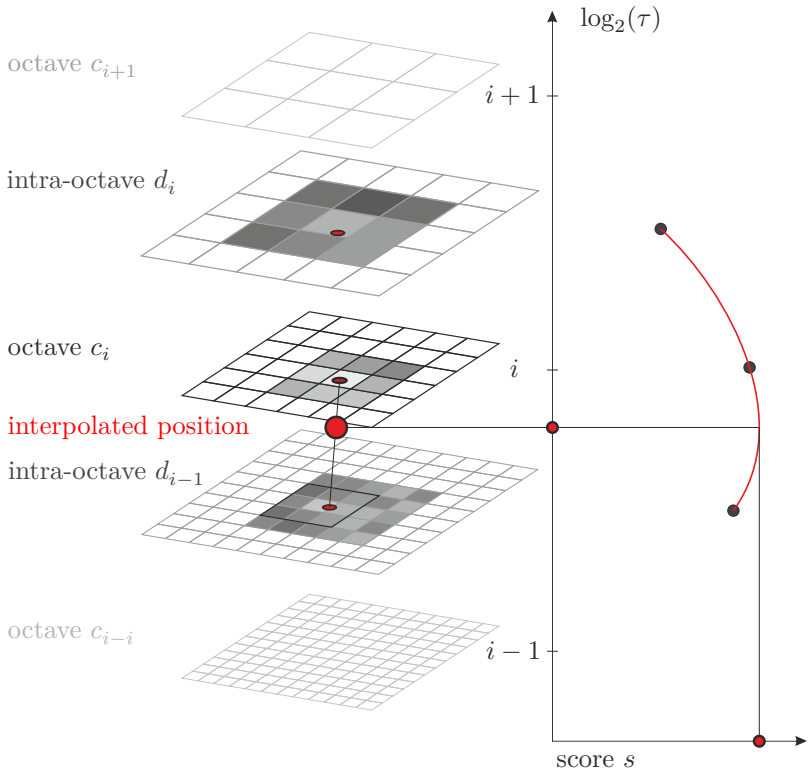
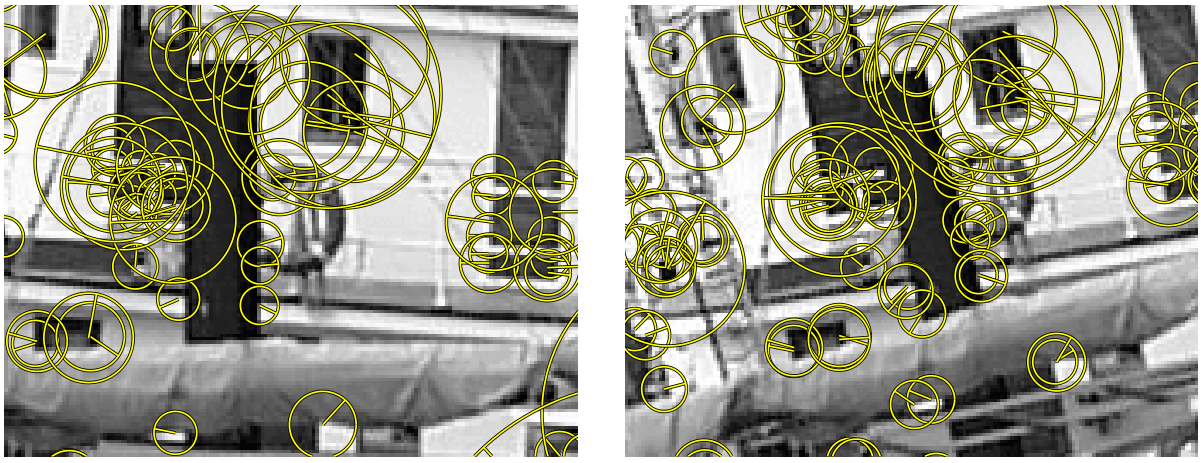


Figure 6.2: Scale-space interest point detection: a keypoint (i.e. saliency maximum) is identified at octave c_i by analyzing the 8 neighboring saliency scores in c_i as well as in the corresponding scores-patches in the immediately-neighboring layers above and below. In all three layers of interest, the local saliency maximum is sub-pixel refined before a 1D parabola is fitted along the scale-axis to determine the true scale of the keypoint. The location of the keypoint is then also re-interpolated between the patch maxima closest to the determined scale.

Initially, the detector, be it FAST/AGAST or Harris, is applied on each octave and intra-octave separately using the same threshold T to identify potential regions of interest. Next, the points belonging to these regions are subjected to a non-maxima suppression in scale-space: firstly, the point in question needs to fulfill the maximum condition with respect to its 8 neighboring scores s in the same layer. The score s is defined as the maximum threshold still considering an image point a corner. Secondly, the scores in

the layer above and below will need to be lower as well. We check inside equally sized square patches: the side-length is chosen to be 2 pixels in the layer with the suspected maximum. Since the neighboring layers (and therefore its FAST scores) are represented with a different discretization, some interpolation is applied at the boundaries of the patch. Figure 6.2 depicts an example of this sampling and the maxima search.

Considering image saliency as a continuous quantity not only across the image but also along the scale dimension, we perform a sub-pixel and continuous scale refinement for each detected maximum. In order to limit complexity of the refinement process, we first fit a 2D quadratic function in the least-squares sense to each of the three scores-patches (as obtained in the layer of the keypoint, the one above, and the one below) resulting in three sub-pixel refined saliency maxima. In order to avoid resampling, we consider a 3 by 3 score patch on each layer. Next, these refined scores are used to fit a 1D parabola along the scale axis yielding the final score estimate and scale estimate at its maximum. As a final step, we re-interpolate the image coordinates between the patches in the layers next to the determined scale. An example of the Harris-based BRISK detection in two images of the Boat sequence (defined in Section 6.4) is shown up-close in Figure 6.3.



(a) Boat image 1. (b) Boat image 2.

Figure 6.3: Close-up of a BRISK detection example using the AGAST/FAST corner score on images 1 and 2 of the Boat sequence exhibiting small zoom and in-plane rotation. The size of the circles denote the scale of the detected keypoints while the radials denote their orientation. For clarity, the detection threshold is set here to a stricter value than in the typical setup, yielding slightly lower repeatability.

6.2.3 Uniformity Enforcement

A specific application of BRISK, namely an efficient and accurate vision-aided odometry or SLAM pipeline relies on tracked keypoints that are uniformly distributed in the image. Optimally, we would like to achieve a target keypoint density associated with a desired number of points per image, while still providing high repeatability. We therefore propose a detection scheme here that favors uniformity by gradually suppressing weaker keypoints in close proximity to strong ones. As opposed to a grid-based strategy that enforces a certain number of keypoints per image-cell, our approach does not suffer from repeatability loss due to change of grid-cell membership of a keypoint between images.

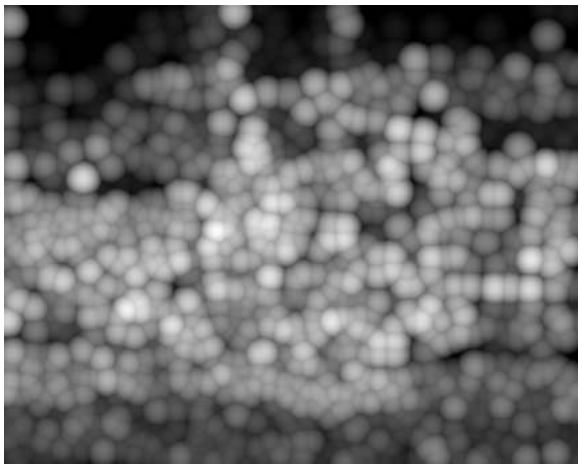
In short, our approach first detects *all* corner score maxima and sorts the respective keypoints by score. Starting with the strongest keypoint, its neighborhood is assigned

an occupancy index proportional to the score decaying with distance to the detected point. At a user-settable distance R and beyond, the index will be zero. This is stored as an occupancy image employing SSE instructions in the implementation. Any successive detection is then compared to the occupancy index and rejected, if its score falls below the stored index at its position. This method allows for keypoints of equal strength to still co-exist, if they are spaced minimally; and it will still detect very weak keypoints in areas that are of low contrast.

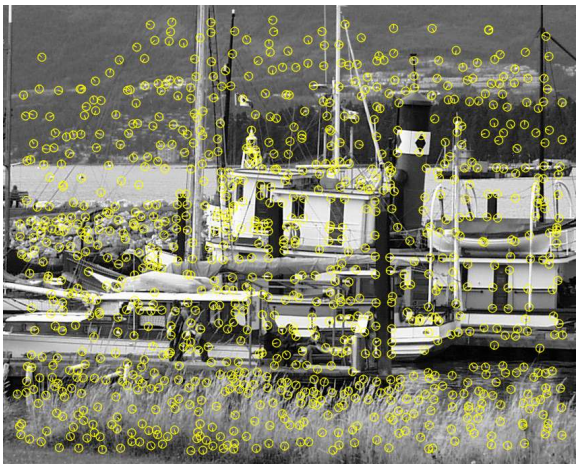
Figure 6.4 shows a comparison of applying the uniformity enforcement step to the standard approach with an absolute cornerness threshold, and also displays the occupancy index image. For simplicity, we display the output of the single-scale version. We used



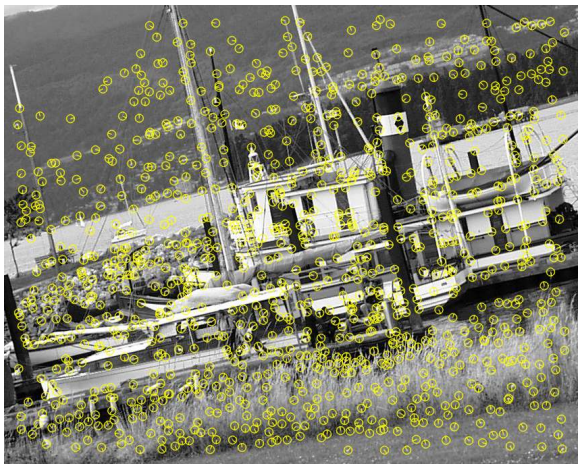
(a) Boat 1 classic (thresholded) detections.



(b) Boat 1 occupancy index image.



(c) Boat 1 with uniformity enforcement.



(d) Boat 2 with uniformity enforcement.

Figure 6.4: Harris-based detection of keypoints in image 1 of the Boat dataset without uniformity enforcement (T =800'000) is shown in 6.4a, occupancy index image as generated during uniformity enforcement in 6.4b. 6.4c and 6.4d display the result with uniformity enforcement ($R = 30$) and also qualitatively assess the validity of the proposed approach with respect to preservation of repeatability. All thresholds were selected to yield approximately 1'000 detections.

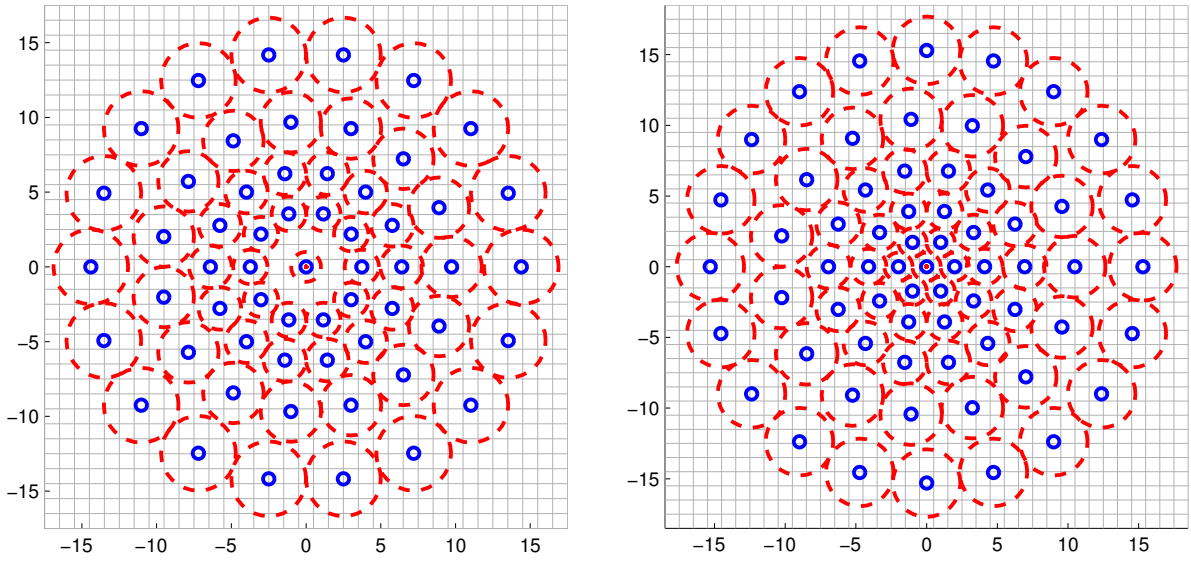
thresholds to output approximately 1'000 keypoints in the example; note that the the number of detections will be much more constant for different images and scenes when using the uniformity-enforced version as opposed to simple thresholding that just considers absolute contrast.

6.3 Rotation Invariance and Descriptor Extraction

Given a set of keypoints (consisting of sub-pixel refined image locations and associated floating-point scale values), the BRISK descriptor is composed as a binary string by concatenating the results of simple brightness comparison tests. This idea has been demonstrated in [Calonder et al. 2010] to be very efficient. Here, however, we employ it in a far more qualitative manner. In BRISK, we identify the characteristic direction of each keypoint to allow for orientation-normalized descriptors and hence achieve rotation invariance which is key to general robustness. Also, we carefully select the brightness comparisons with the focus on maximizing descriptiveness.

6.3.1 Sampling Pattern and Rotation Estimation

The key concept of the BRISK descriptor makes use of a pattern applied for sampling the neighborhood of the keypoint. The pattern, illustrated in Figure 6.5, defines N locations equally spaced on circles concentric with the keypoint. We show both the



(a) BRISK pattern version 1.

(b) BRISK2 pattern.

Figure 6.5: The BRISK sampling pattern with $N = 60$ points in the first version (left) and the modified second version of BRISK2 with $N = 66$ points (right). The small blue circles denote the sampling locations; the bigger, red dashed circles are drawn at a radius σ corresponding to the standard deviation of the Gaussian kernel used to smooth the intensity values at the sampling points. The pattern shown applies to a scale of $\tau = 1$.

original version as published in [Leutenegger et al. 2011], as well as an enhanced version abbreviated BRISK2. While this pattern resembles the DAISY descriptor [Tola et al. 2010], it is important to note that its use in BRISK is entirely different, as DAISY was built specifically for dense matching, deliberately capturing more information and thus resulting in demanding speed and storage requirements.

In order to avoid aliasing effects when sampling the image intensity of a point $\mathbf{p}_i$ in the pattern, we apply Gaussian smoothing with standard deviation σ_i proportional to the distance between the points on the respective circle. Positioning and scaling the pattern

accordingly for a particular keypoint k in the image, let us consider one of the $N(N-1)/2$ sampling-point pairs $(\mathbf{p}_i, \mathbf{p}_j)$. The smoothed intensity values at these points which are $I(\mathbf{p}_i, \sigma_i)$ and $I(\mathbf{p}_j, \sigma_j)$ respectively, are used to estimate the local gradient $\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j)$ by

$$\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I(\mathbf{p}_j, \sigma_j) - I(\mathbf{p}_i, \sigma_i)}{\left\|\mathbf{p}_j - \mathbf{p}_i\right\|^2}. \tag{6.1}$$

Considering the set $\mathcal{A}$ of all sampling-point pairs:

$$\mathcal{A} = \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\right\} \tag{6.2}$$

we define a subset of short-distance pairings $\mathcal{S}$ and another subset of L long-distance pairings $\mathcal{L}$:

$$\begin{aligned} \mathcal{S} &= \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \left\|\mathbf{p}_j - \mathbf{p}_i\right\| < \delta_{\mathcal{S}}\right\} \subseteq \mathcal{A}, \\ \mathcal{L} &= \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \left\|\mathbf{p}_j - \mathbf{p}_i\right\| > \delta_{\mathcal{L}}\right\} \subseteq \mathcal{A}. \end{aligned} \tag{6.3}$$

As part of the BRISK2 modifications, we in fact apply an alternative approach to obtaining the pairings $\mathcal{S}_2$ and $\mathcal{L}_2$. The idea behind the change was normalizing distances locally with the spatial extent of the smoothing:

$$\begin{aligned} \mathcal{S}_2 &= \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \left\|\mathbf{p}_j - \mathbf{p}_i\right\| - \sigma_j - \sigma_i < (\sigma_j + \sigma_i)\delta_{\mathcal{S}_2}\right\} \subseteq \mathcal{A}, \\ \mathcal{L}_2 &= \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \left\|\mathbf{p}_j - \mathbf{p}_i\right\| - \sigma_j - \sigma_i < (\sigma_j + \sigma_i)\delta_{\mathcal{L}_2}\right\} \subseteq \mathcal{A}. \end{aligned} \tag{6.4}$$

In words, we weigh the thresholds for short and long pairings with the smoothing kernel sizes and include also shorter pairings in $\mathcal{L}_2$. The threshold distances in the original BRISK are set to $\delta_{\mathcal{S}} = 9.75\tau$ and $\delta_{\mathcal{L}} = 13.67\tau$ (τ is the scale of k). In BRISK2, we set $\delta_{\mathcal{S}_2} = 1.16$ and $\delta_{\mathcal{L}_2} = 2.17$. Iterating through the point pairs in $\mathcal{L}$ (or $\mathcal{L}_2$ in BRISK2), we estimate the overall characteristic pattern direction of the keypoint k to be:

$$\mathbf{g} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{L}} \mathbf{g}(\mathbf{p}_i, \mathbf{p}_j). \tag{6.5}$$

6.3.2 Building the Descriptor

For the formation of the rotation- and scale-normalized descriptor, BRISK applies the sampling pattern rotated by $\alpha = \arctan2(g_y, g_x)$ around the keypoint k. The bit-vector descriptor d_k is assembled by performing all the short-distance intensity comparisons of point pairs $(\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$ (i.e. in the rotated pattern), such that each bit b corresponds to:

$$b = \left\{ \begin{array}{ll} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{array} \right\}, \forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}. \tag{6.6}$$

While the BRIEF descriptor is also assembled via brightness comparisons, BRISK has some fundamental differences apart from the obvious pre-scaling and pre-rotation of the sampling pattern. Firstly, BRISK uses a deterministic sampling pattern resulting in a uniform sampling-point density at a given radius around the keypoint. Consequently, the tailored Gaussian smoothing will not accidentally distort the information content of a brightness comparison by blurring two close sampling-points in a comparison.

Furthermore, BRISK uses dramatically fewer sampling-points than pairwise comparisons (i.e. a single point participates in multiple comparisons), limiting the complexity of looking up intensity values. Finally, the comparisons here are restricted spatially such that the brightness variations are only required to be locally consistent. With the sampling pattern and the distance thresholds as shown above, we obtain a bit-string of length 512 in the original BRISK version. The bit-string of BRIEF64 also contains 512 bits, thus the matching for a descriptor pair will be performed equally fast by definition. In BRISK2, however, the descriptor is only composed of to a 384 bits, thus reducing matching time and storage by 25%.

6.3.3 Descriptor Matching

Matching two BRISK descriptors is a simple computation of their Hamming distance as done in BRIEF [Calonder et al. 2010]: the number of bits different in the two descriptors is a measure of their dissimilarity. Notice that the respective operations reduce to a bitwise XOR followed by a bit count, which can both be computed very efficiently on today's architectures.

6.3.4 Notes on Implementation

Here, we give a very brief overview of some implementation issues which contribute significantly to the overall computational performance and the reproducibility of the method. All the BRISK functionality builds on the common 2D feature interface of OpenCV 2.2 and newer allowing easy integration and interchangeability with existing features (SIFT, SURF, BRIEF, etc.).

The detection process uses the AGAST implementation [Mair et al. 2010] for computing saliency scores. The non-maxima suppression benefits from early termination capability limiting the saliency scores calculation to a minimum. Building the image pyramid makes use of some SSE2 and SSSE3 commands, both concerning the halfsampling as well as the downsampling by a factor of 1.5.

In order to efficiently retrieve gray values with the sampling pattern, we generate a look-up table of discrete rotated and scaled BRISK pattern versions (consisting of the sampling-point locations and the properties of the Gaussian smoothing kernel as well as the indexing of long and short distance pairings) consuming around 40 MB of RAM—which is typically still acceptable for applications constrained to little available memory. As part of the BRISK2 modifications, we in fact only store the unrotated pattern locations, which significantly reduces storage and pattern generation time, at low expense of computation.

We furthermore use the integral image along with a simplified Gaussian kernel version inspired by [Bay et al. 2008]: the kernel is scalable when changing σ without any increase in computational complexity. In our final implementation, we use as an approximation a simple square box mean filter with floating point boundaries and side length $\rho = 2.6\sigma$.

Thus we do not need time-consuming Gaussian smoothing of the whole image with many different kernels, but we instead retrieve single values using an arbitrary parameter σ.

We also integrated an improved SSE Hamming distance calculator achieving matching at 6 times the speed of the current OpenCV implementation as used for example with BRIEF in OpenCV.

6.4 Performance Evaluation

Our proposed method has been extensively tested following the now established evaluation method and datasets in the field first proposed Mikolajczyk and Schmid [2005]; Mikolajczyk et al. [2005]. For the sake of consistency with results presented in other works, we also used their MATLAB evaluation scripts which are available online. Each of the datasets contains a sequence of six images exhibiting an increasing amount of transformation. All comparisons here are performed against the first image in each dataset. Figure 6.6 shows one image for each dataset analyzed.



(a) Graffiti.



(b) Wall.



(c) Boat.



(d) Ubc.



(e) Leuven.



(f) Bikes.



(g) Trees.

Figure 6.6: Datasets used for evaluation: viewpoint change (Graffiti and Wall), zoom and rotation (Boat), JPEG compression (Ubc), brightness change (Leuven), and blur (Bikes and Trees).

The transformations cover view-point change (Graffiti and Wall), zoom and rotation (Boat), blur (Bikes and Trees), brightness changes (Leuven) as well as JPEG compression (Ubc). Since the viewpoint change scenes are planar, the image pairs in all sequences are provided with a ground truth homography used to determine the corresponding keypoints. In the rest of the section, we present quantitative results concerning the detector and descriptor performance of BRISK compared to SIFT (OpenCV2.4 implementation) as well as SURF (version 1.0.9[3]). Our evaluation uses similarity matching which considers any pair of keypoints with descriptor distance below a certain threshold a match—in contrast to e.g. nearest neighbor matching, where a database is searched for the match with the lowest descriptor distance. Finally, we also demonstrate BRISK's big advantage in computational speed by listing comparative timings.

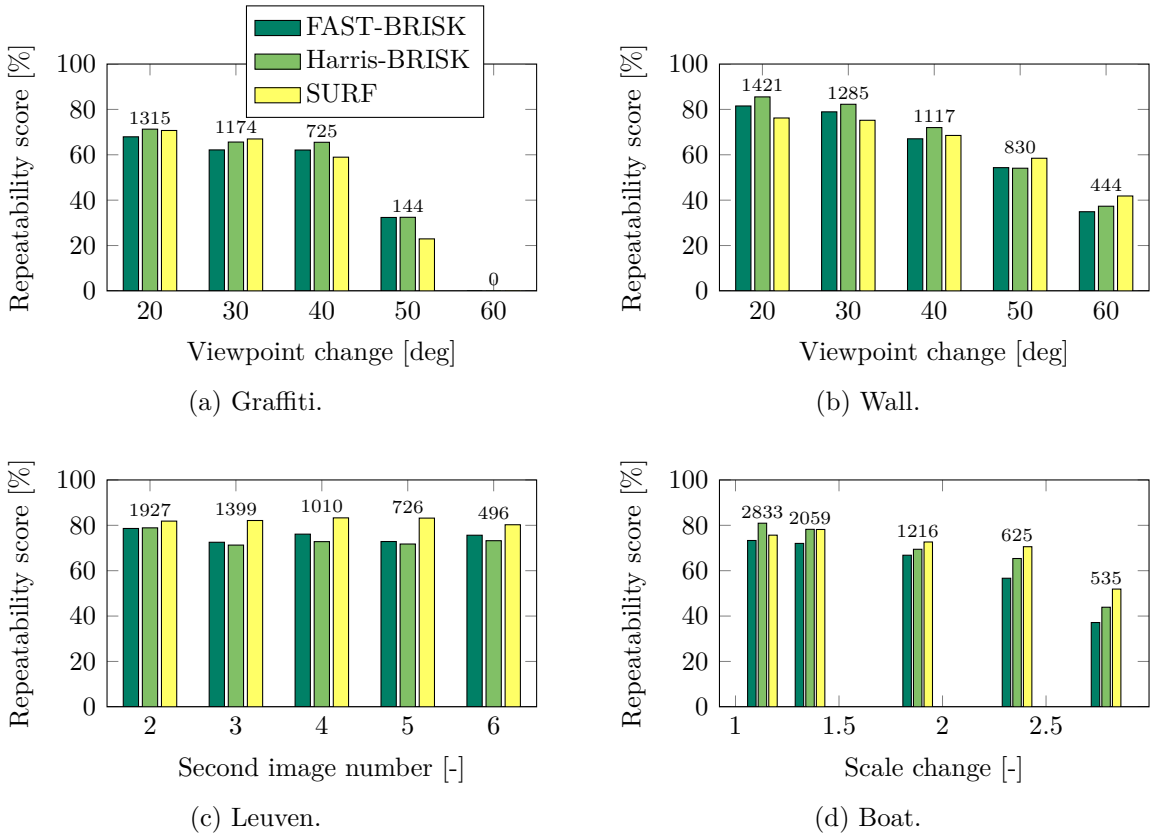
6.4.1 Detection Repeatability

The detector repeatability score as defined in [Mikolajczyk et al. 2005] is calculated as the ratio between the corresponding keypoints and the minimum total number of keypoints visible in both images. The correspondences are identified by looking at the overlap area of the keypoint region in one image (i.e. the extracted circle) and the projection of the keypoint region from the other image (i.e. ellipse-like): if the region of intersection is

[3]SURF V1.0.9. available at www.vision.ee.ethz.ch/ surf, as of May 2014.

larger than 50% of the union of the two regions, it is considered a correspondence. Note that this method is largely dependent on the assignment of the keypoint circle radius, i.e. the constant factor between scale and radius. We choose this such that the average radii obtained with the BRISK detector approximately match the average radii obtained with the SURF and SIFT detectors.

The assessment of repeatability scores (a selection of results is shown in Figure 6.7) is performed using constant BRISK detection thresholds across one sequence. For the



(a) Graffiti. (b) Wall.

(c) Leuven. (d) Boat.

Figure 6.7: Repeatability scores for 50% overlap error of the FAST-BRISK, Harris-BRISK and the SURF detector. The similarity correspondences of Harris-BRISK (approximately matched between the detectors) are given as numbers above the bars.

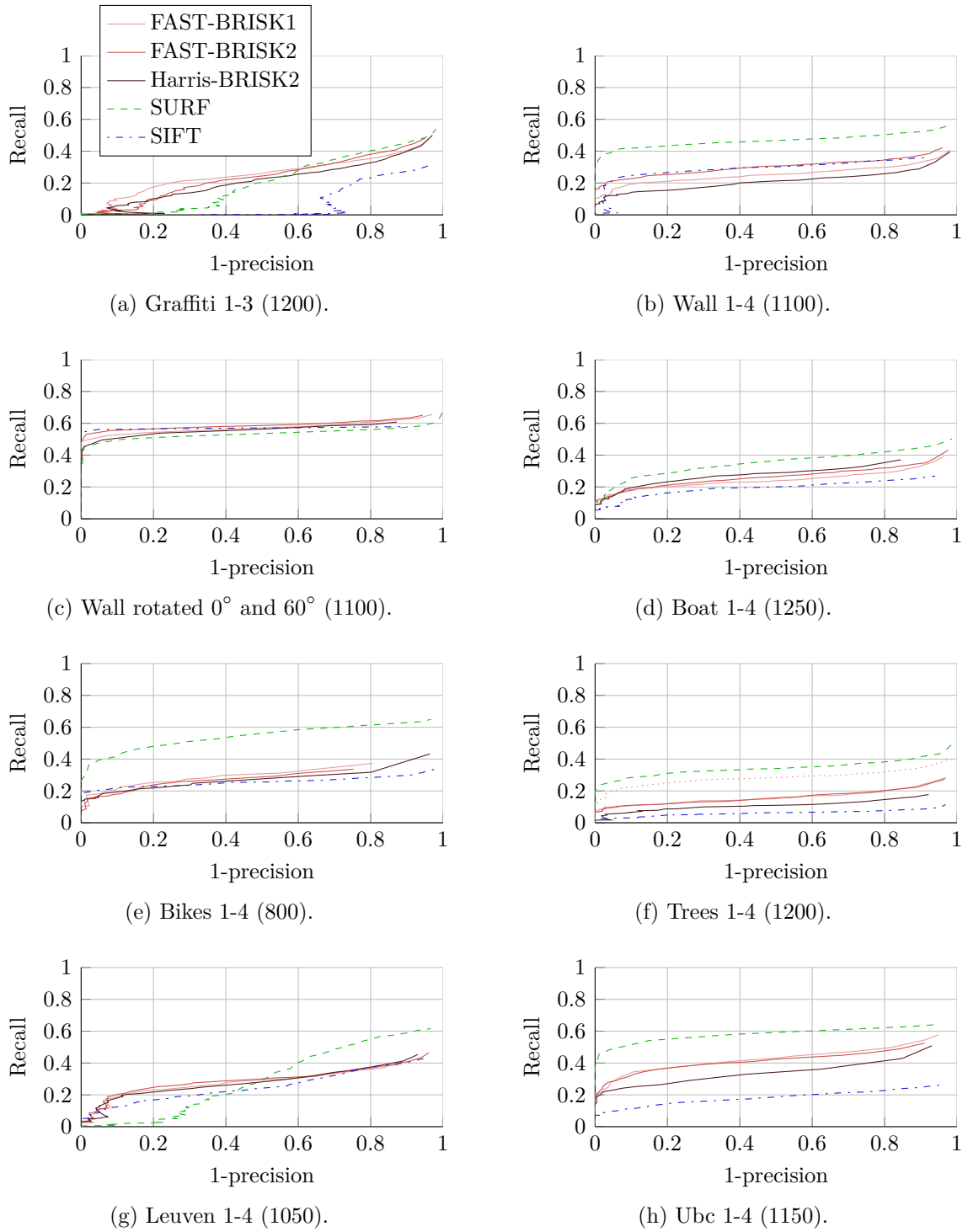
sake of a fair comparison with the SURF detector, we adapt the respective Hessian threshold such that it outputs approximately the same number of correspondences in the similarity based matching setup. Clearly, both BRISK detector versions exhibit equivalent repeatability as the SURF detector as long as the image transformations applied are moderate; the Harris-based version even outperforms SURF in some datasets. Given the clear advantage in computational cost of the BRISK over the SURF detector, however, the proposed method constitutes a strong competitor, even if the performance at larger transformations appears to be slightly inferior.

6.4.2 Overall Algorithms Descriptor Similarity

Since our work aims at providing an overall fast as well as robust detection, description and matching, we evaluate the joint performance of all these stages in BRISK and compare it to SIFT and SURF.

With Scale and Rotation Invariance

Figure 6.8 shows the precision-recall curves using threshold-based similarity matching for a selection of image pairs of different datasets.



(a) Graffiti 1-3 (1200).

(b) Wall 1-4 (1100).

(c) Wall rotated 0° and 60° (1100).

(d) Boat 1-4 (1250).

(e) Bikes 1-4 (800).

(f) Trees 1-4 (1200).

(g) Leuven 1-4 (1050).

(h) Ubc 1-4 (1150).

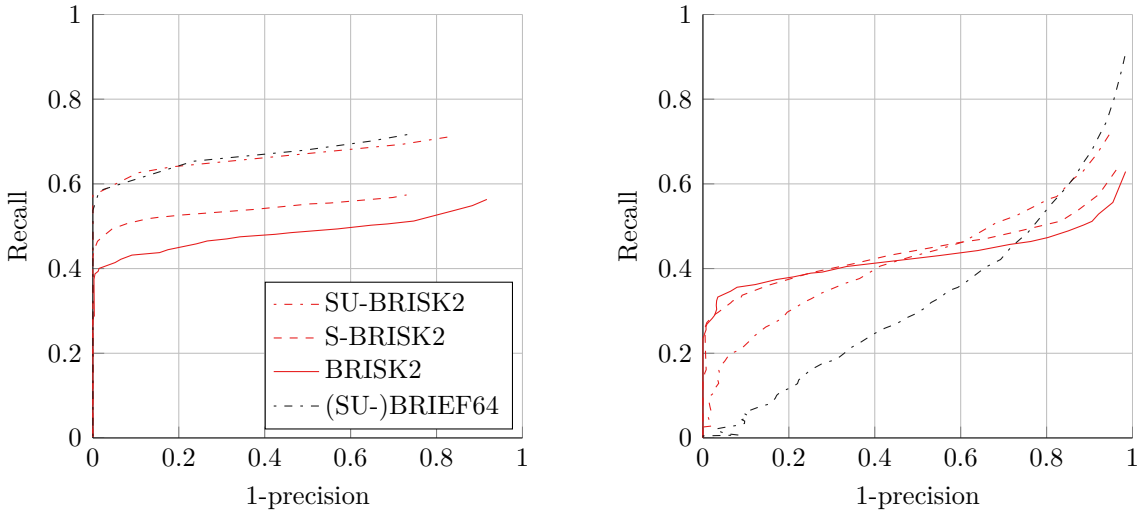
Figure 6.8: Evaluation results for all detection, extraction and matching stages jointly: precision-recall curves of for BRISK, SURF and SIFT. Results are shown for viewpoint changes (a and b), pure in-plane rotation (c), zoom and rotation (d), blur (e and f), brightness changes (g) and JPEG compression (h). The numbers of similarity correspondences are indicated in perentheses. In trees, we provide also BRISK descriptors from SURF regions (dotted).

Again, for this assessment, we adapt the detection thresholds such that all detectors output an approximately equal number of correspondences in the spirit of fairness. Note that the evaluation results here are different from the ones in [Bay et al. 2006], where all descriptors are extracted on the same regions (obtained with the Fast-Hessian detector).

As illustrated, BRISK performs competitively with SIFT and SURF in all datasets and even outperforms the other two in some cases. The reduced performance of BRISK in the Trees dataset is attributed to the detector performance: while SURF detects 1260 and 1372 regions in the images, respectively, BRISK only detects 827 regions in image 4 compared to 2707 found in image 1 to achieve the approximate same number of correspondences. The same holds for the other blur dataset, Bikes: saliency as assessed with FAST or Harris is inherently more sensitive to blur than blob-like detectors. We therefore also show the evaluation of the BRISK descriptors extracted from the SURF regions for the Trees dataset, demonstrating again that the descriptor performance is comparable to SURF. Evidently, SIFT performs significantly worse in the Trees, Boat, and Ubc datasets, which can be explained with the limited detector repeatability in these cases. On the other hand, SIFT and BRISK handle the important case of pure in-plane rotation well.

The Use of Invariances and Comparisons to BRIEF and FREAK

Furthermore, we would like to highlight situations, in which invariances are advantageous, and others, where they deteriorate system performance. We make the link to BRIEF, which was designed to be neither rotation nor scale invariant. Figure 6.9 shows a comparison of the unrotated, single-scale BRISK version (SU-BRISK) to 64 byte BRIEF features on the same (single scale) AGAST/FAST keypoints. Also included are the rotation invariant, single-scale S-BRISK, as well as the standard BRISK keypoints. The

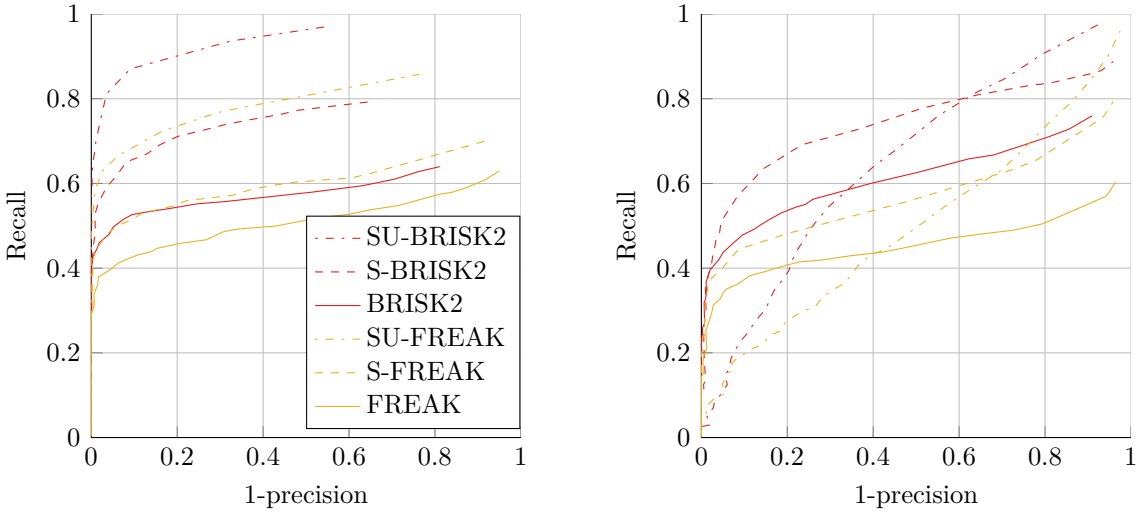


(a) Wall 1-2 (1000). (b) Boat 1-2 (1600).

Figure 6.9: Comparison of different BRISK2-versions to 64 byte BRIEF. BRIEF, as well as both SU-BRISK2 (single-scale, unrotated) and S-BRISK (single-scale) are extracted from AGAST keypoints detected in the original image. Notice that the BRISK pattern was scaled such that it matches the BRIEF patch size. The standard version of BRISK had to be extracted from our scale-invariant corner detection with adapted threshold to match the number of correspondences (given in brackets).

experiment is conducted with two image pairs: on the one hand, we used the first two images in the Wall dataset proving that SU-BRISK and BRIEF64 exhibit a very similar performance in the absence of scale change and in-plane rotation. Notice that this situation is an example of the intended application domain of BRIEF. On the other hand, we applied the different versions to the first two images of the Boat sequence: this experiment demonstrates some advantage of the SU-BRISK over BRIEF in terms of robustness against small rotation (10°) and scale changes (10%). The well-known and intuitive price for both rotation and scale invariance is easily observable.

As a further comparison, we also provide plots that include a derivative of BRISK, called "FREAK: Fast Retina Keypoints" [Alahi et al. 2012]. Note that FREAK by default uses a *much* larger keypoint neighbourhood for sampling brightness than BRISK. Thus, in order to obtain a fair comparison, we scaled the BRISK pattern to the size consumed by FREAK. The results shown in Figure 6.10 clearly demonstrate the superior



(a) Wall 1-2 (900).

(b) Boat 1-2 (1500).

Figure 6.10: FREAK versus BRISK2 (with pattern scaled to match FREAK): BRISK2 clearly outperforms FREAK, despite the 25% shorter descriptor.

performance of BRISK2 over FREAK, despite the fact that BRISK2 only uses 384 bits compared to 512 in the FREAK descriptor. When comparing to Figure 6.9, the massive influence of the sampling pattern scale becomes evident.

6.4.3 Timings

Timings were recorded on a laptop with a quad-core Intel® Core™ i7-2675QM 2.20GHz processor with disabled turbo boost (only using one core, however) running Ubuntu 12.04 (64-bit), using the implementation and setup as detailed above. Table 6.1 presents the results concerning detection on the first image of the Graffiti sequence, while Table 6.2 shows the matching times. The values are averaged over 100 runs. Note that all matchers do a brute-force descriptor distance computation without any early termination optimizations.

The timings show a clear advantage of BRISK. Its detection and descriptor computation is typically an order of magnitude faster than the one of SURF, which are

	SIFT	SURF	FAST-BRISK2
Detection threshold	3.9	64000	83
Number of points	1189	1255	708
Detection time [ms]	262.7	122.6	16.71
Description time [ms]	251.7	631.3	5.32
Total time [ms]	514.5	753.9	22.03
Time per point [ms]	0.4327	0.6007	0.03112

Table 6.1: Detection as well as extraction timings for the first image in the Graffiti sequence (size: 800×640 pixels).

	SIFT	SURF	BRISK2
Points in first image	1189	1229	708
Points in second image	1712	1504	979
Total time [ms]	107.6	107.2	11.23
Time per comparison [ns]	52.85	58.01	16.20

Table 6.2: Matching timings for the Graffiti image 1 and 3 setup.

considered to be the fastest rotation and scale invariant features currently available. It is also important to highlight that BRISK is easily scalable for faster execution by reducing the number of sampling-points in the pattern at some expense of matching quality—which might be affordable in a particular application. Moreover, scale and/or rotation invariance can be omitted trivially, increasing the speed as well as the matching quality in applications where they are not needed.

6.5 Discussion

We have presented a novel method dubbed BRISK, which tackles the classic Computer Vision problem of detecting, describing and matching image keypoints for cases without sufficient *a priori* knowledge on the scene and camera poses. In contrast to well-established algorithms with proven high performance, such as SIFT and SURF, the method at hand offers a dramatically faster alternative at comparable matching performance—a statement which we base on an extensive evaluation using an established framework. BRISK relies on an easily configurable circular sampling pattern from which it computes brightness comparisons to form a binary descriptor string. The unique properties of BRISK can be useful for a wide spectrum of applications, in particular for tasks with hard real-time constraints or limited computation power: BRISK finally offers the characteristics of high-end features in such time-demanding applications.

We provide both a FAST/AGAST score based scale-space corner detector, as well as a Harris-based version. The experiments have shown the competitiveness with SIFT and SURF in terms of descriptor similarity, of both our original BRISK as well as an enhanced version, BRISK2, the descriptor of which was reduced from 512 bits to 384. We furthermore presented and evaluated methodologies of enforcing a homogeneous keypoint distribution in the image as desired by vision-aided odometry.

Amongst avenues for further research into BRISK, we aim to further explore alternatives to the scale-space maxima search of saliency scores to yield higher repeatability whilst maintaining speed. Furthermore, we aim at analyzing both theoretically and experimentally the BRISK pattern and the configuration of comparisons, such that the information content and/or robustness of the descriptor is maximized.

Chapter 7
Autonomous Aircraft Flight Testing

Flight testing of with Unmanned Aerial Systems (UAS) finally reveals the applicability of the aircraft, as it has been designed from a conceptual stage to detail design, from aerodynamics to electronics subsystems, algorithms and their actual software implementations. We provide an overview of flight operations performed with our UAS, focusing on solar airplane testing. Specifically, an overview of impressions and lessons learned is given, followed by results of the various state estimators proposed in the previous chapters.

7.1 Introduction



(a) senseSoar pre-flight checks. (b) senseSoar in flight.

Figure 7.1: senseSoar operations on the airfield of Hausen am Albis: only after extensive pre-flight checks an actual flight can be performed. Photographs by François Pomerleau.

Field testing is on the one hand very rewarding to the roboticists, when a system demonstrates to be applicable in a real environment; on the other hand, unmanned

Parts of this chapter appear in:

- S. Leutenegger, A. Melzer, K. Alexis, and R. Y. Siegwart. Robust State Estimation for Small Unmanned Airplanes. In *Systems and Control (MSC), 2014 IEEE International Multi-Conference on*. IEEE, 2014, submitted.

aircraft as presented in this work are highly complex systems, a single point of failure of which can lead to potentially catastrophic events. In this respect, we will summarize our experience in Section 7.2—a first impression of which is shown in Figure 7.1. Minor incidences have happened during the test series, and also a major one, namely a crash with the senseSoar airplane that lead to significant damage: the suspected causes and lessons learned will be provided as well.

In a second part, Section 7.3, results of the estimators presented in Chapter 5 will be provided as applied to flight data from our UAS. We will analyze the applicability of the filtering-based Inertial Navigation System (INS) to the senseSoar airplane. Furthermore, we will show the visual-inertial localization and mapping algorithm that we deployed on multicopters as well as on the sensor pod mounted to AtlantikSolar and augmented with GPS and magnetometer measurements.

7.2 Impressions, Challenges and Lessons Learned

Field testing is teamwork. None of the presented results below would have been achieved without the help of a large group of very skilled, motivated and hard-working Autonomous Systems Lab (ASL) members. Furthermore, successful flights start much before the day and time of take-off. Below, we provide a compilation of factors we regard as key to effective and safe testing of unmanned solar airplanes.

- **Redundancy and safety mechanisms.** Of most importance are safety mechanisms in terms of automatic control versus manual Remote Control (RC) to be implemented in hardware and software: our final configuration involves a multiplexer that allows hardware-level switching to conventional RC model mode, so there exists no dependency on an operational autopilot board. Conversely, if the system detects RC signal loss, it will switch to automatic mode and loiter, until the connection is resumed.

- **Component testing in the lab.** Testing of all the subsystems individually and then altogether in *exactly* the flight configuration has shown to be an absolute necessity; any kind of interference or compatibility problem can only be identified by this procedure. The availability of Hardware-In-the-Loop (HIL) functionality has further accelerated our module testing, in particular in terms of autopilot implementation.

- **Testing sites.** The space requirements on the ground as well as in the air are considerable, thus limiting the suitable places near the lab. We have been testing on five different sites so far. We are furthermore in the process of certifying our aircraft for flight within segregated airspace that we applied for allocation above testing sites, so we may perform operations beyond Line of Sight (LoS) in future.

- **Non-solar test airplanes.** Performing test flights with solar airplane prototypes is a very resource demanding undertaking, due to their complexity, weight and size that ask for tedious preparation and dedicated testing sites. Having access to small and simple test platforms like the EasyGlider[2] for quick testing near the lab has

[2] Available from www.multiplex-rc.de, accessed May 2014

sped up our testing cycles dramatically, since crashes can be easily recovered from (in the worst case by replacing the airframe).

- **Weather assessment.** Some background knowledge is required in terms of weather forecast, in order to select appropriate time slots and test sites such that operations can be performed safely (mostly related to wind and precipitation).

- **Test protocol.** A clear protocol of the operations to be tested needs to be established in advance that is agreed upon.

- **Check-lists.** Pre-flight, in-flight and post-flight checklists covering normal and emergency operations have been established and have to be followed rigorously, in order to ensure safe operations.

- **Ground station.** Testing can only be performed in an efficient and deterministic way, if sufficient introspection during flight is provided. To this end, we employ an adapted version of QGroundControl[3].

- **Field equipment.** Tools and introspection instruments are of crucial importance on the field. Figure 7.2 illustrates the classic setup prior to a test flight: we mount the sensing and processing pod to AtlantikSolar (see Chapter 3). Pre-flight checks have been performed. In the background, the ground station including antenna for telemetry were set up.

- **Analysis tools.** A suite of easy-to-use tools for fast analysis, post-processing and visualization of logged flight data concerning estimator, controller, and system state have to be available.



(a) AtlantikSolar ready after pre-flight checks.

(b) AtlantikSolar test site setup.

Figure 7.2: AtlantikSolar test site in Rafz (ZH): airplane and pilot (Rainer Lotz) ready with RC (left); and ASL crew mounting the sensing and processing unit, with ground station and telemetry antenna in the background (right).

[3]QGroundControl open-source project, see www.qgroundcontrol.org, accessed May 2014.

7.2.1 Analysis of the senseSoar Crash

Despite the experience and procedures as overviewed above we had already accumulated, a serious crash of the senseSoar airplane has happened near Russikon (ZH) on December 15th 2013.

Prior to the crash, we had successfully performed the first hand-launch of senseSoar, and subsequently tuned the autopilot parameters of the Stability Augmentation System (SAS) as well as Control Augmentation System (CAS, attitude control). We then asked the pilot to fimiliarize again with the purely manual RC mode for five minutes, in order to perform the landing without any autopilot action. After one missed approach because of too high altitude, the pilot lost control over the aircraft in a turn during the second approach—after which the airplane entered a steep spiraling motion and hit the ground at high speed and pitch angle.

We used our analysis pipeline to draw conclusions on the causes. Figure 7.3 displays the senseSoar path before the crash and Figure 7.4 shows the final 10 seconds of the most relevant logged on-board INS filter outputs (described in Section 5.4). Around

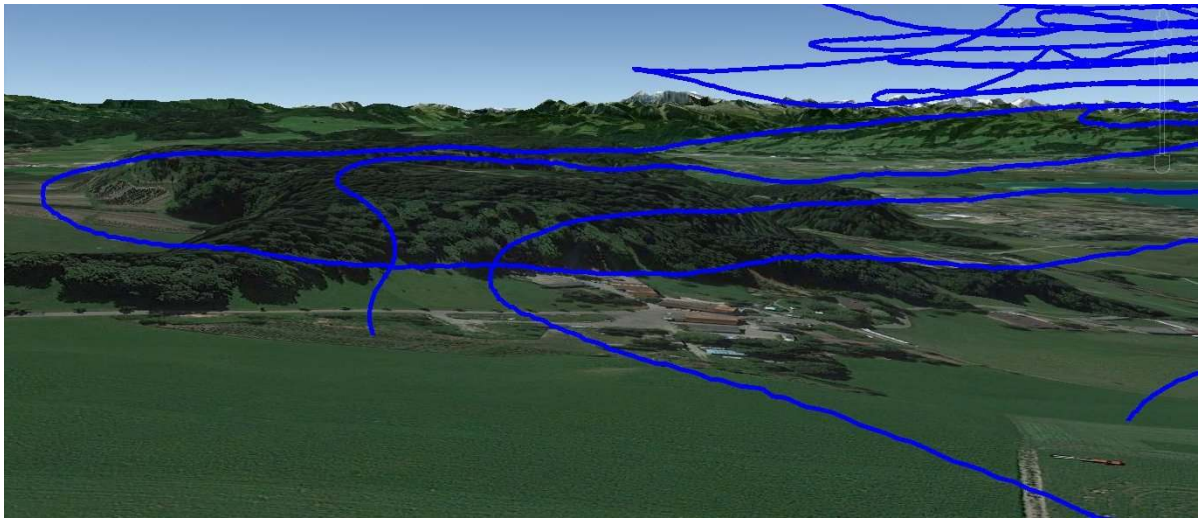


Figure 7.3: senseSoar path before crash visualized in google earth. The spin turns are clearly visible in the middle left part of the image.

timestamp 4828 s, a right turn is initiated and develops to 50° of bank angle. At the same time, the angle of attack rises and shows a peak of 11° at timestamp 4830.6 s, where stall is suspected. In the following, the pitch angle reaches up to 60 to 90° nose down quickly, along with an acceleration in the turning motion: it looks like a classic spin. At timestamp 4833.3 s, the airplane hits the ground at around 45° pitch down. Due to some sensor failures, the reconstruction after the crash is unusable.

At this point, we cannot exclude an actuator failure to be responsible for the steep turn, although it seems improbable. An RC failure, however, is highly unlikely, since the RC signal was logged without interruptions. Despite serious structural damage, the airframe was repaired. In order to avoid similar future incidences, we will consider the following lessons learned:

- Flying in SAS and CAS mode (deliberately) changes the dynamics of the aircraft. An extended re-familiarization phase to manual mode with specific maneuvers to

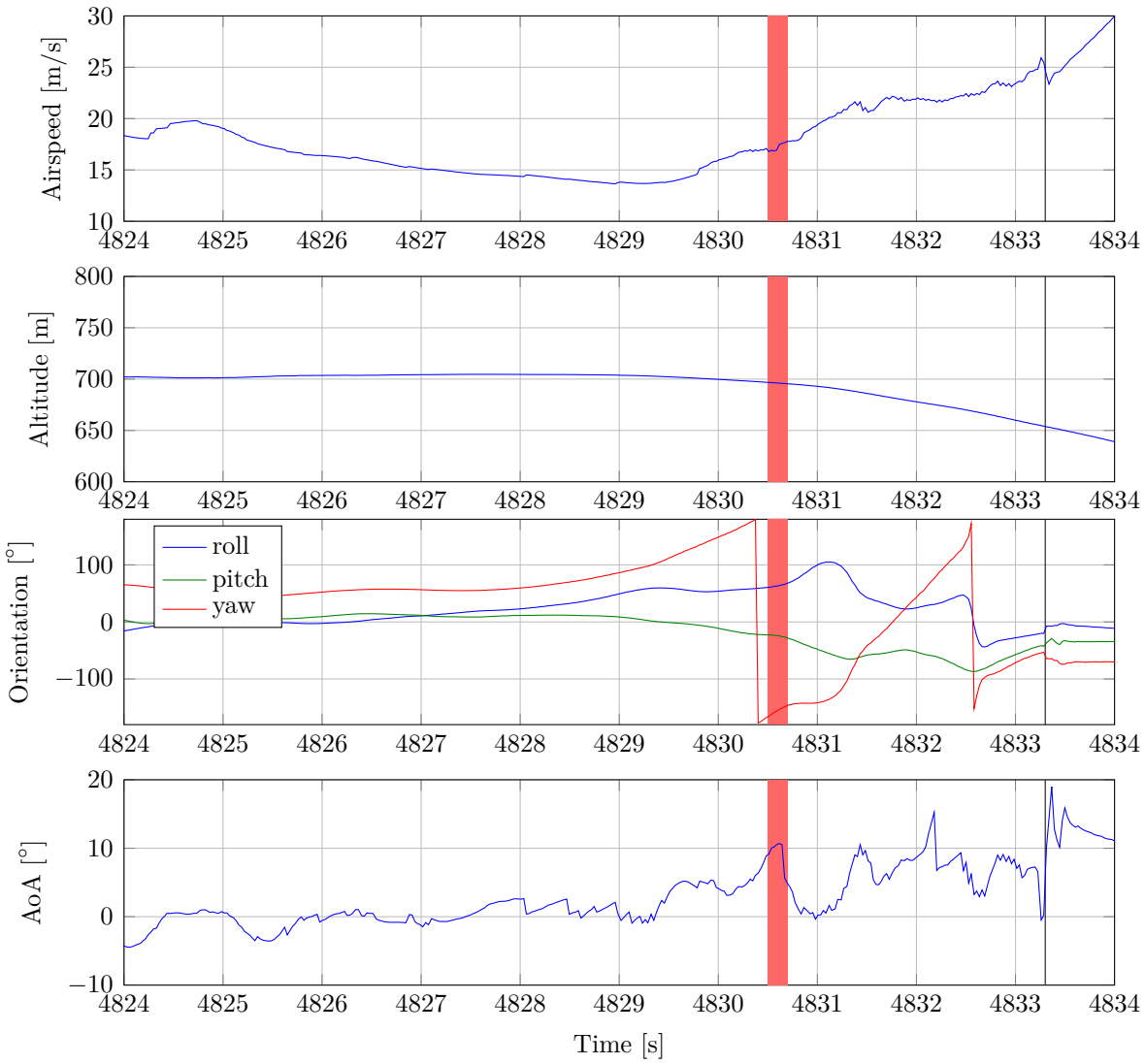


Figure 7.4: senseSoar crash online state estimates: in a steep turn, the stall Angle of Attack (AoA) is reached at timestamp 4830.6 s (wide red vertical bar), followed by a spin into the ground that is hit at timestamp 4833.3 s (black vertical bar).

be flown at reasonably high altitudes above ground is paramountly recommended.

- The airplane's behaviour at high angles of attack is more aggressive than previously assumed. Although the Center of Gravity (CoG) was located within the limits, a more forward position will be targeted in future. Furthermore, the yaw directional stability and control authority is rather limited. Design modifications are to be elaborated.

- Turns of more than 30° bank angle with solar airplanes operating at low speeds are not recommended—certainly not at low altitude.

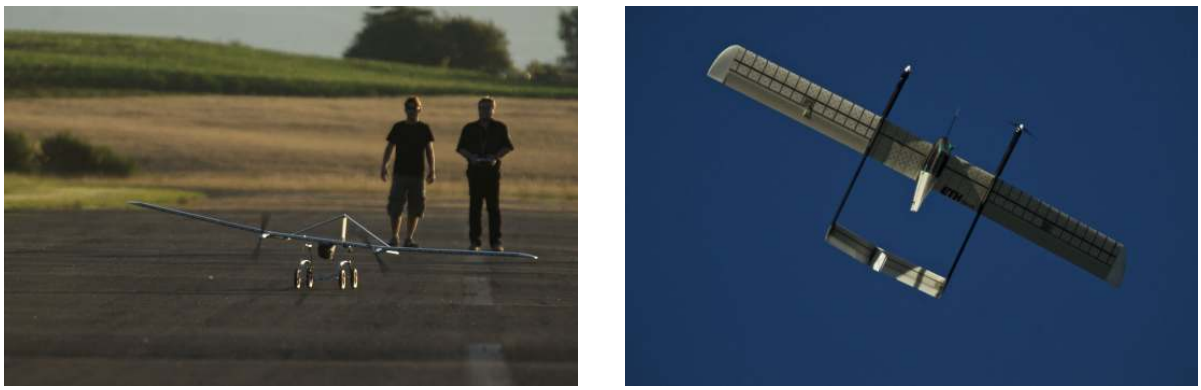
- An acoustic stall warning via the ground station is highly recommended. This, however, requires the pilot to stand close enough to the ground station, which is not always possible. The event finally also underlines that automatic stall prevention and recovery should be implemented as part of the autopilot.

7.3 State Estimation Results

Below, we show three use-cases of the state estimation algorithms described in 5: the INS filter presented in Section 5.4 that has been run online as an input to the autopilot on many platforms is analyzed in terms of robustness first. Next, we will show the application of the visual-inertial estimation described in Section 5.5 to run on-board a multicopter system—in its stereo version and in a GPS-denied setting. Finally, we show the application of the monocular visual-inertial version of the algorithm on an AtlantikSolar dataset, where we also included magnetometer and GPS measurements into the optimization.

7.3.1 Multi-Sensor Estimation in the Control Loop

The INS filter that does not use imagery was applied to manned glider flight and proved to be applicable also during extended GPS outages (see Section 5.4.2). Furthermore, it was given the capability of detecting and ignoring outliers in some of the sensor measurements. The filter has been used as input to the autopilot system of all ASL fixed wing UAS for many hours. Here, we analyze its performance when processing sensor measurements from the senseSoar airplane. We use the Specifically, we will show results of three cases: we start with regular operation, then analyze outlier handling and finally look at GPS loss. The aerodynamic airplane polars describing vertical and sideslip forces are used as part of the estimator, enabling Angle of Attack (AoA) and sideslip angle tracking. The flight was recorded at Hausen am Albis airfield for senseSoar system identification; thus in parts, longitudinal and lateral dynamics are highly excited. The take-off was performed from a starting cart on the runway as illustrated in Figure 7.5



(a) senseSoar take-off from cart. (b) senseSoar airborne.

Figure 7.5: senseSoar take-off and flight on Hausen am Albis airfield: for some of the first test flights, we used a cart and took off from the runway. Photographs by François Pomerleau.

Table 7.1 lists the parameters related to the sensors employed. Here, the px4-internal sensors were used, i.e. rather low-end MEMS components without factory calibration or temperature compensation[4].

[4]The sensors are listed at pixhawk.org/modules/px4fmu, accessed May 2014.

Characteristic	Symbol	Value	Unit
Gyro noise density	$\sigma_{\mathrm{g,c}}$	1.7e-4	$\mathrm{rad}/(\mathrm{s}\sqrt{\mathrm{Hz}})$
Gyro drift noise density	$\sigma_{\mathrm{b_g,c}}$	1.0e-4	$\mathrm{rad}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Acc. noise density	$\sigma_{\mathrm{a,c}}$	0.05	$\mathrm{m}/(\mathrm{s}^2\sqrt{\mathrm{Hz}})$
Acc. drift noise density	$\sigma_{\mathrm{b_a,c}}$	0.0022	$\mathrm{m}/(\mathrm{s}^3\sqrt{\mathrm{Hz}})$
Acc. time constant	τ	18000	s
Wind drift noise density	$\sigma_{\mathrm{b_a,c}}$	0.5	$\mathrm{m}/(\mathrm{s}\sqrt{\mathrm{Hz}})$
GPS position noise NED	$\boldsymbol{\sigma}_{\mathrm{p}}$	[2.2,2.2,10]	m
Magnetometer noise	σ_{m}	1	μT
Static pressure noise	σ_{s}	10.0	Pa
Dynamic pressure noise	σ_{d}	2.0	Pa

Table 7.1: senseSoar sensor characteristics.

Normal Operation

Figure 7.6 shows the flight path visualized in Google Earth. In addition, we display

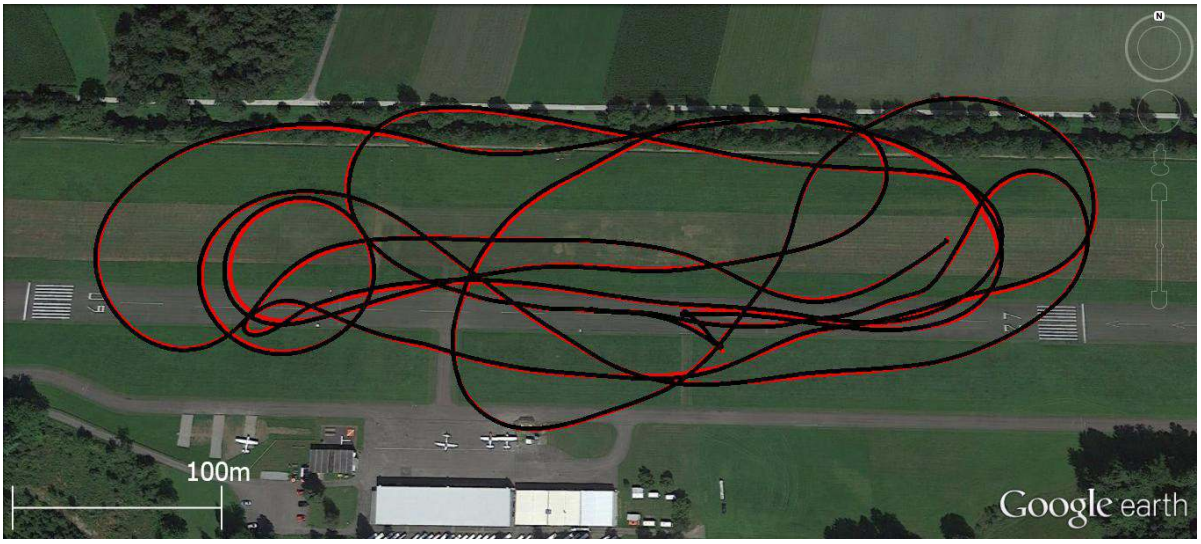


Figure 7.6: Google Earth overhead plot of the GPS (black) and position estimate path (red) in normal operation.

the orientation states in Figure 7.7. Note the excitation on roll and yaw angles in some parts. The rest of the states is summarized in Figure 7.8. Note the large bias values of the low-cost MEMS sensors that are, however, tracked in a stable way. As derived from the state vector, we also compute the AoA and sideslip angle, visualized in Figure 7.9. Note the sideslip oscillations in two sections which is a consequence of rudder excitation. The wind is estimated with a significant variance, which can be explained in parts with actual fluctuations of the wind (turbulence), but probably also with the unusually excited dynamics of the airplane during that flight. As typical for such experiments, ground truth for many of the states is unfortunately not available, most notably for AoA and orientation.

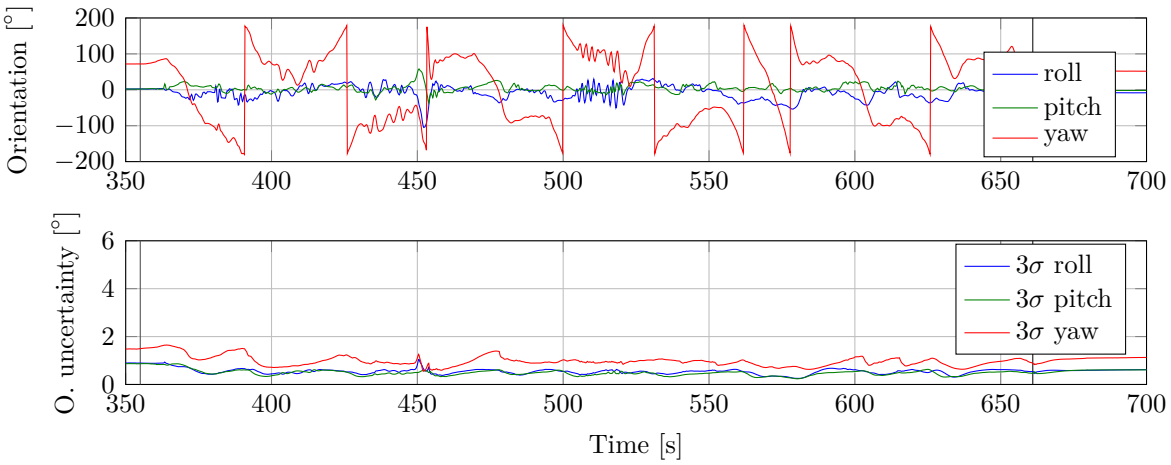


Figure 7.7: Orientation estimates with 3σ-bounds in normal operation. Parts of significant roll and yaw excitation are clearly visible in the middle of the flight. Vertical bars: take-off and landing.

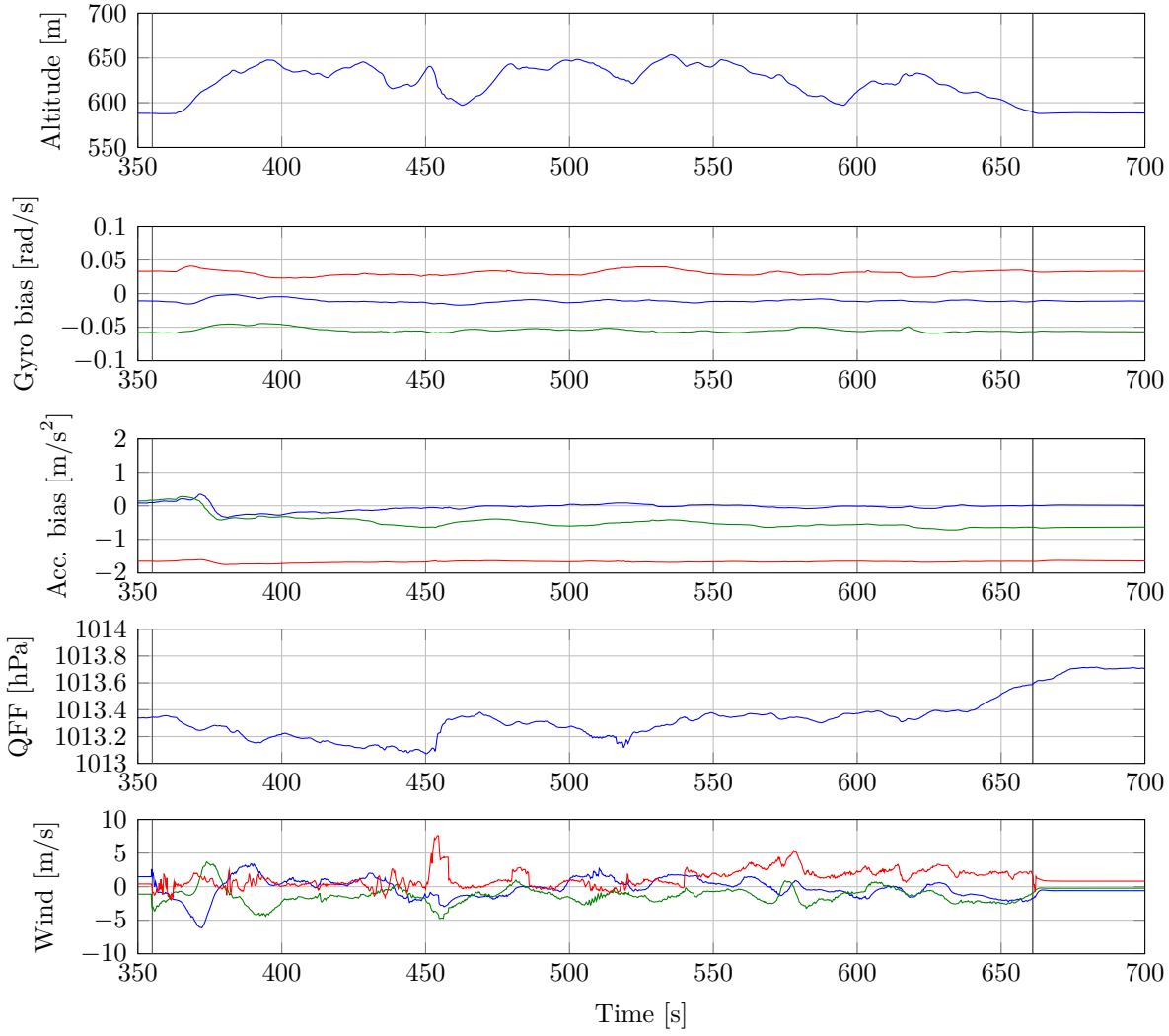


Figure 7.8: Remaining estimated states in normal operation. Red, green, blue denote x, y, and z components, where applicable.

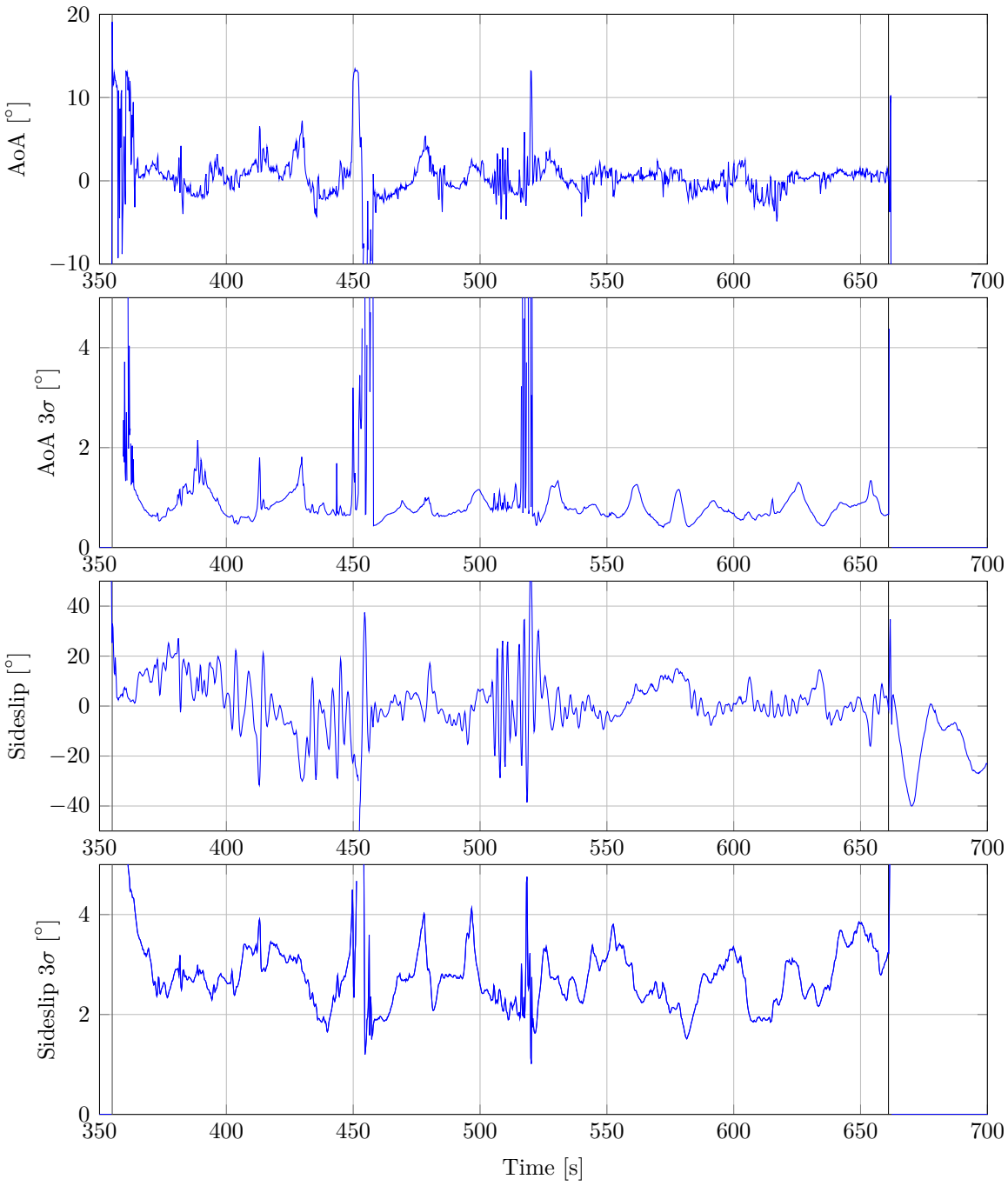


Figure 7.9: Angle of Attack (AoA) and sideslip angles with 3σ-bounds in normal operation. Clearly and expectedly, the high excitations on the sideslip angle cause the AoA and sideslip angles to be inaccurate for the periods of concern, which shows also in the uncertainty estimation.

Outlier Handling

Outliers as caused by hardware or firmware issues are supposed to happen very rarely, but they have a potentially catastrophic effect on the estimator and all succeeding functionality. In order to underline the validity of our chosen approach based on the χ^2 residual error (see Section 5.4), we corrupt the pressure and magnetometer signal as visualized in Figure 7.10 and feed the estimator with them.

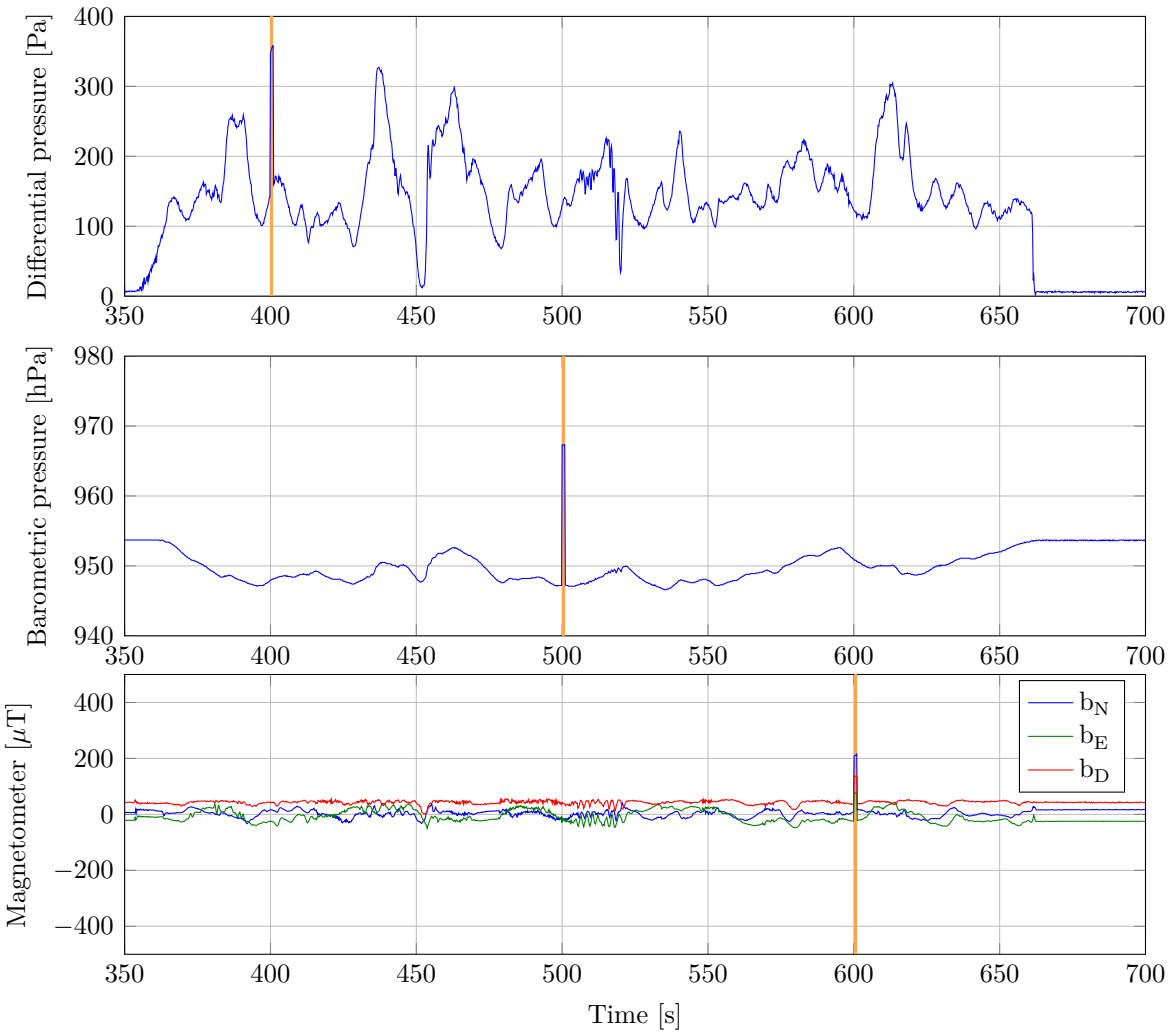


Figure 7.10: Corrupted data: one second of outlier peaks were inserted at $t = 400$ seconds for the differential pressure, at $t = 500$ seconds for the static pressure, and at $t = 600$ seconds for all magnetometer components.

The impact on orientation is shown in Figure 7.11. The differences of the remaining

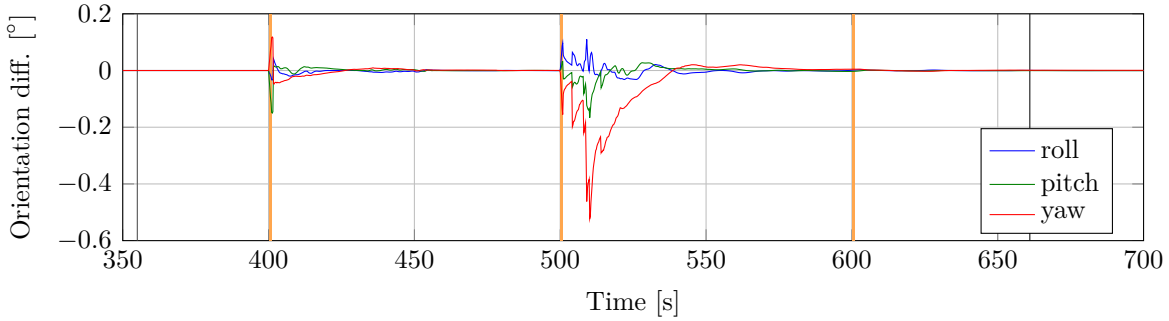


Figure 7.11: Orientation estimates difference in presence of outliers.

states are essentially very small. AoA and sideslip, for instance, deviate shortly in the order of 1° and 2°. Clearly, the outliers only have a small impact on the estimation process.

GPS Outage

We simulate a GPS outage of 100 seconds, starting at $t = 552.5$ seconds. Of most interest are the characteristics of the orientation estimation in this case as compared to the normal operation, which we visualize in Figure 7.12. While roll and pitch angle, which

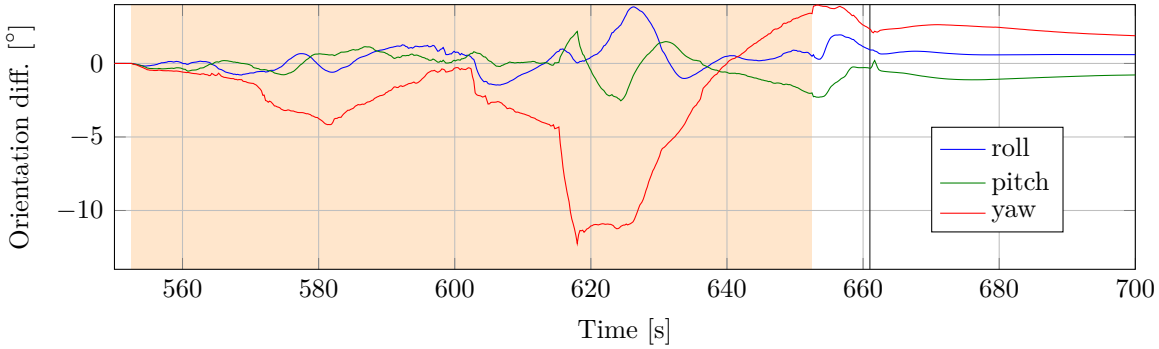


Figure 7.12: Difference of orientation estimates under GPS outage conditions from $t = 552.5$ seconds to $t = 652.5$ seconds.

are most relevant for automatic control, show an RMS error in the order of one degree, the yaw partly deviates further. AoA and sideslip are compared to the normal operations baseline in Figure 7.13. Sideslip estimates deviate further as expected, due to limited

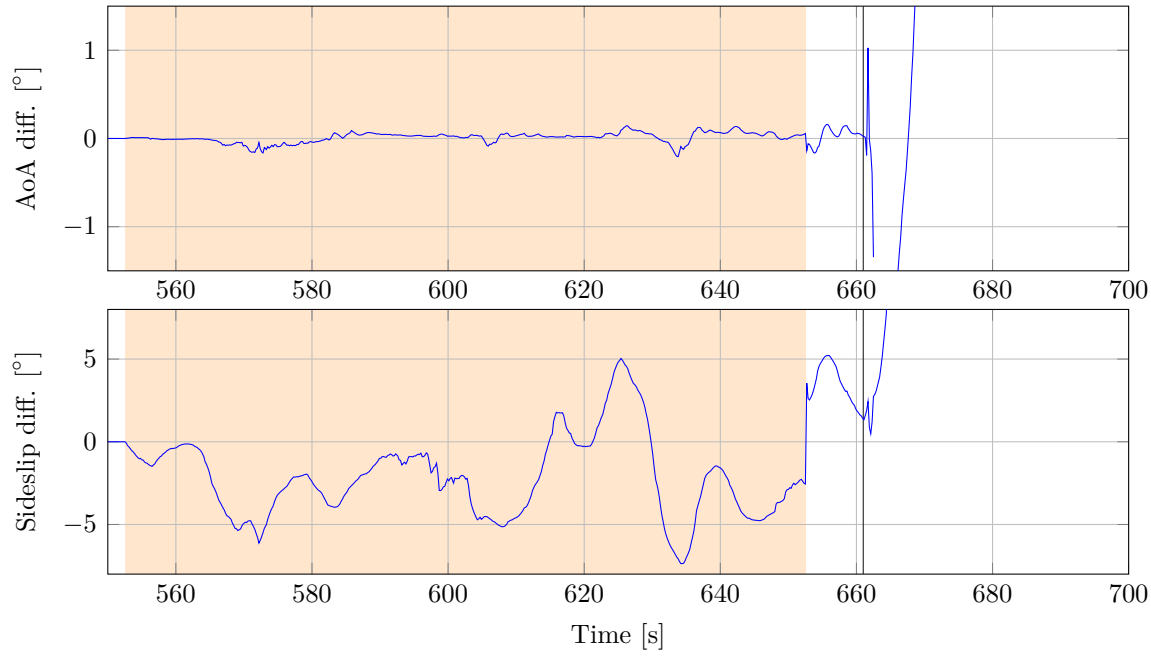


Figure 7.13: Differences of Angle of Attack (AoA) and sideslip angles under GPS outage conditions compared to normal operation. Note the landing at $t = 662$ s.

observability via small acceleration and an inaccurate aerodynamics model of the sideslip force.

While not relevant for airplane stabilization, we still analyze how the position drifts when GPS measurements are missing: the overhead plot is shown in Figure 7.14, where a position drift during outage is clearly visible as expected. In order to save space, we again omit to show the rest of the states.

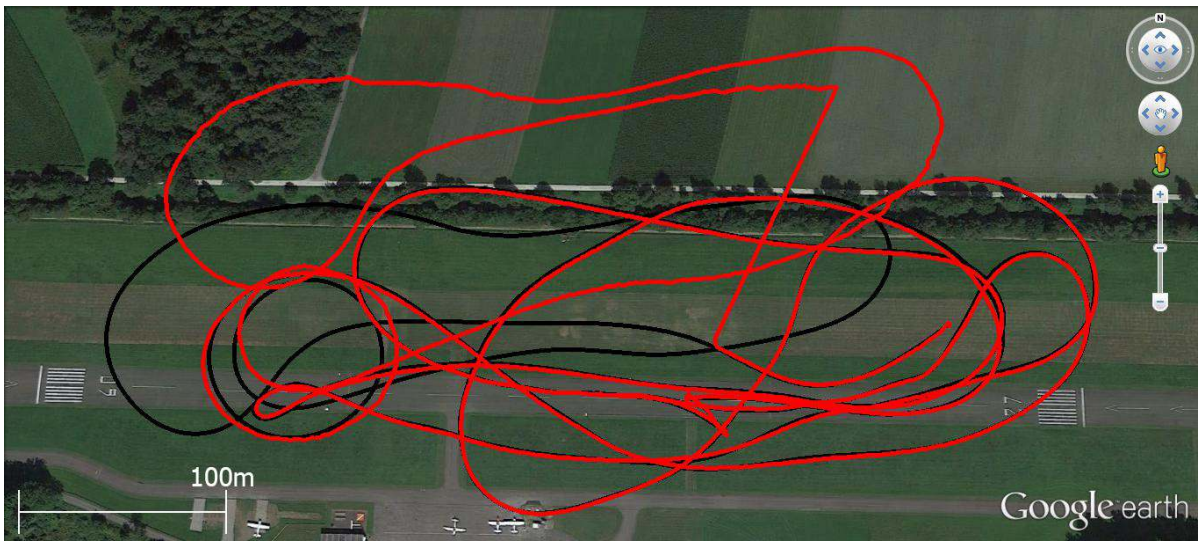


Figure 7.14: Google Earth overhead plot of the GPS (black) and position estimate path (red) under simulated GPS outage conditions.

7.3.2 Stereo-Visual-Inertial Odometry On-Board Multicopter

We have ported the keyframe based visual-inertial odometry outlined in Section 5.5 to run on-board a Firefly Hexacoptor[5] to which the ASL/Skybotix visual-inertial sensor[6] was mounted. Processing takes place on the on-board Intel® Core™ 2 Duo computer. The platform is shown in Figure 7.15.

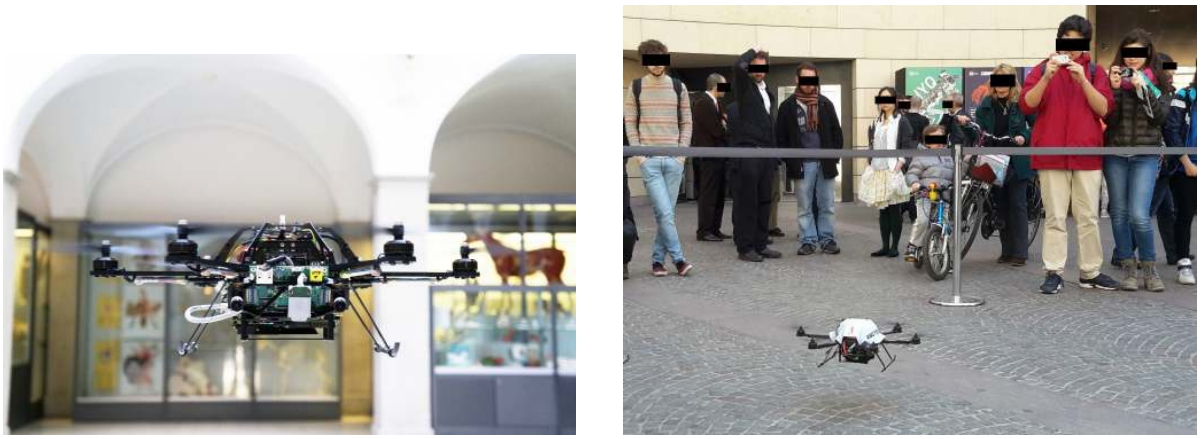


Figure 7.15: Hexacopter with stereo-visual-inertial sensor.

The respective pose estimate is used as input to the controller. We have demonstrated automatic operation with this setup at several public and research related demonstrations, including RSS 2013 in Berlin, the European Robotics Forum 2014 in Rovereto, and ICRA

[5] See www.asctec.de, accessed May 2014.
[6] Specifications available at www.skybotix.com, accessed May 2014.

2014 in Hong Kong. Figure 7.16 shows the flying system in action. All flights were



(a) At ETH Zurich (photo: Skybotix). (b) European Robotics Forum in Rovereto 2014.

Figure 7.16: Hexacopter operating with on-board visual-inertial odometry: the output is used for fully automatic control. Note the spectators acting as non-static objects in the scene.

performed without estimator robustness issues, despite challenging environments with variable lighting conditions, natural visual features only and dynamic objects (such as spectators).

Finally, the aforementioned multicopter was also sent into a goldmine as part of an inspection application with an attached flashlight and laser scanner: the visual-inertial pose estimates were used as a basis to project the laser scans, in order to obtain a denser map than simply the sparse point cloud. An impression of the setup where GPS is definitely not an option is given in Figure 7.17.



Figure 7.17: Hexacopter inside a gold mine: the setup was augmented with a flashing light as well as a laser scanner.

7.3.3 Extended Visual-Inertial Fusion On Solar Airplane

Accurate pose estimation and precisely registered imagery form the basis for applications such as surveillance, mapping, disaster scenario monitoring or Search and Rescue (SaR) in general. Furthermore, online implementations will be of crucial importance when targeting terrain following and avoidance, as well as fully automatic landing. Therefore, as a proof of concept towards performing on-board localization and (sparse) mapping with a solar UAS, we mounted the sensor pod to AtlantikSolar (for details see Section 3.3). The actual setup is shown in Figure 7.18.



Figure 7.18: Sensing and processing pod mounted to AtlantikSolar: picture taken after landing.

The setup only uses one camera. In order to achieve highest accuracy and to safely recover the scale and absolute orientation, we therefore complemented the mono-visual inertial version of the algorithm described in Section 5.5 with GPS and magnetometer measurements, and then provide reconstruction results of a dataset processed offline.

Figure 7.19 shows an impression of the aerial imagery and the keypoint matching between current frame and the latest keyframe. The datasets were recorded flights near Rafz (ZH).

7.3.4 Using GPS Measurements

We use a GPS measurement represented in the world frame ${}_W\tilde{\mathbf{r}}_A$ at time t_{GPS}, where A denotes the antenna location. The GPS error term is now formulated as

$$\mathbf{e}_{\text{GPS}} = {}_W\tilde{\mathbf{r}}_A - \left({}_W\mathbf{r}_S(\mathbf{x}_{\text{R}}^k) + \mathbf{C}_{WS}\left(\mathbf{x}_{\text{R}}^k\right){}_S\mathbf{r}_A\right), \tag{7.1}$$

where ${}_W\mathbf{r}_S(\mathbf{x}_{\text{R}}^k)$ and $\mathbf{C}_{WS}\left(\mathbf{x}_{\text{R}}^k\right)$ denote the position and rotation matrix propagated to time t_{GPS} based on the robot state estimate at step k, $\mathbf{x}_{\text{R}}^k$. At this point, we regard this propagation as a simple interpolation scheme between times at which actual estimates

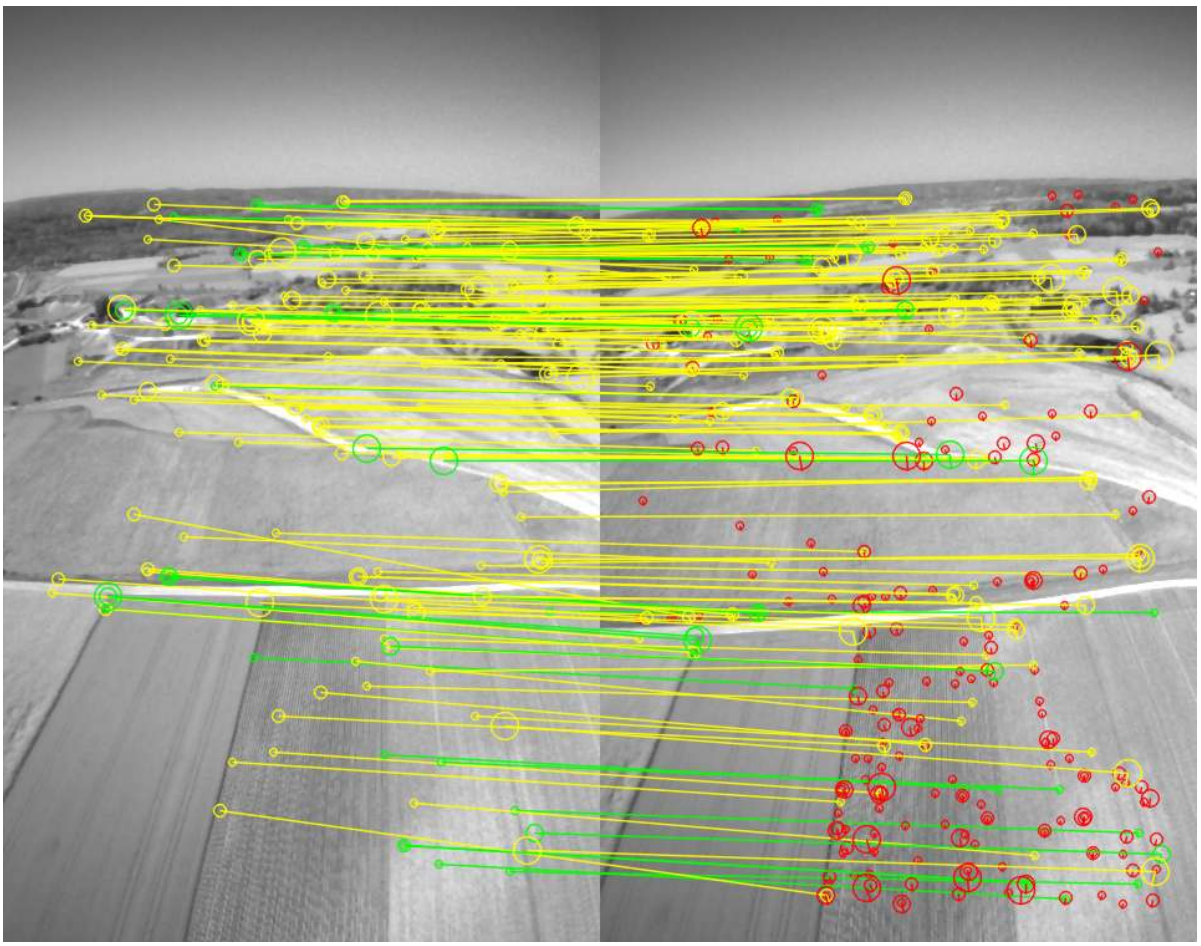


Figure 7.19: AtlantikSolar sensor pod images and keypoint matching: matches of keypoints detected in the current frame (right) to the latest keyframe (left). Green keypoints and match lines indicate initialized depth.

exist, i.e. propagation uncertainty and correlations with IMU error terms are ignored. ${}_S\mathbf{r}_A$ stands for the offset between sensor frame (IMU) $\underrightarrow{\mathcal{F}}_S$ and GPS antenna, represented in $\underrightarrow{\mathcal{F}}_S$. We chose 3 m as lateral measurement standard deviation, and 4 m as vertical standard deviation to weigh the error term.

The Jacobian of this error term is straightforward to derive, but requires the inclusion of the IMU propagation Jacobian as derived in Section 5.5.3.

7.3.5 Inclusion of the Magnetometer

The magnetometer error term used is slightly more sophisticated, in order not to negatively affect the overall algorithm accuracy: we introduce a magnetometer bias $\mathbf{b}_\mathrm{m}$ (hard iron calibration) as a time varying state in the form of a bounded random walk. Furthermore, an offset of the magnetic flux density z-component b_z (in $\underrightarrow{\mathcal{F}}_W$) is used, likewise subject to (bounded) variation over time. We use again the IGRF-11 model (as in Section 5.4) to compute the theoretical magnetic flux density vector ${}_W\mathbf{b}$. With the measurement ${}_S\tilde{\mathbf{b}}$ in

$\underrightarrow{\mathcal{F}}_S$, the error term is expressed as

$$\mathbf{e}_{\mathrm{m}} = {}_W\mathbf{b} - \begin{bmatrix} 0 \\ 0 \\ b_z^k \end{bmatrix} - \mathbf{C}_{WS}\left(\mathbf{x}_{\mathrm{R}}^k\right)\left({}_S\tilde{\mathbf{b}} - \mathbf{b}_{\mathrm{m}}^k\right), \tag{7.2}$$

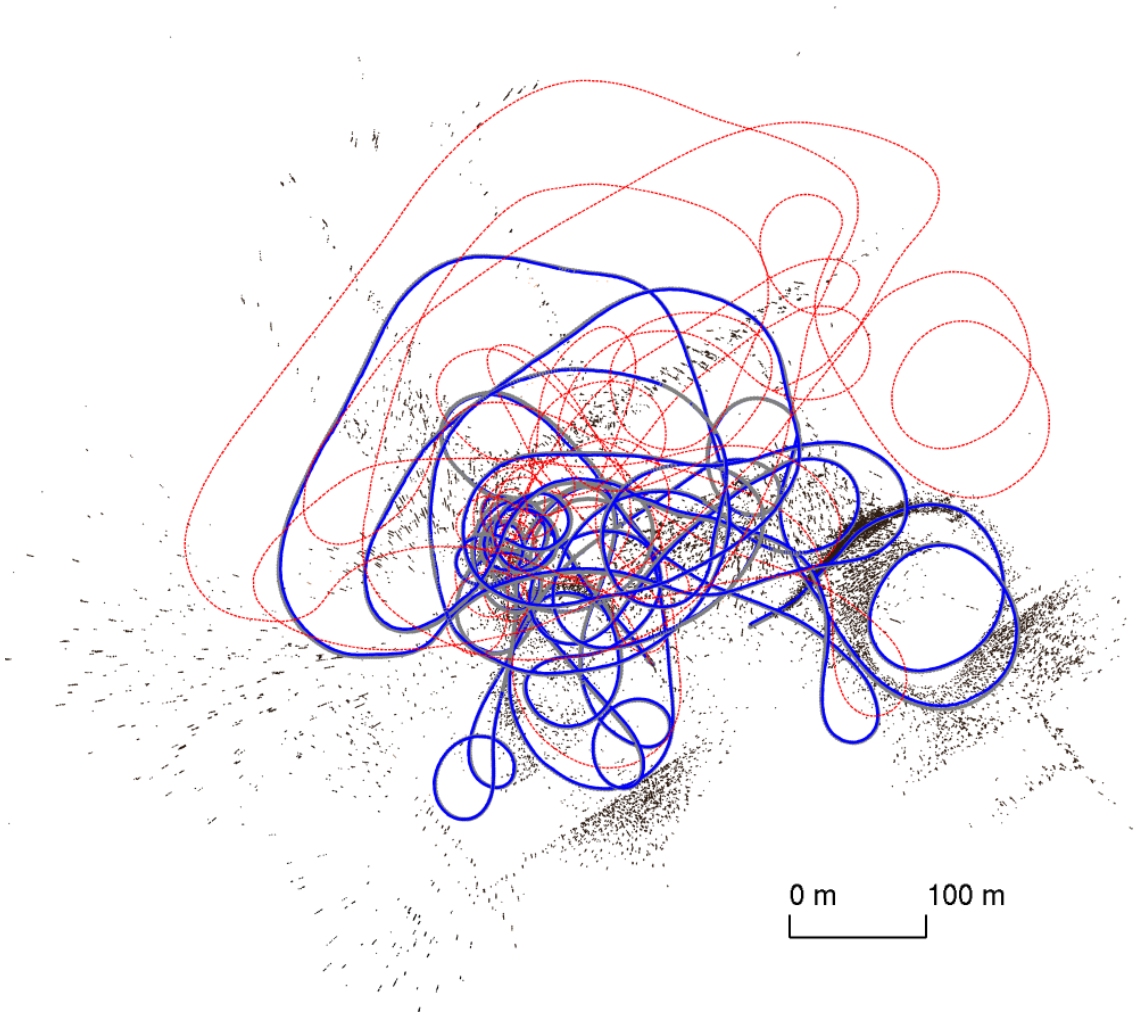
where again the propagated orientation $\mathbf{C}_{WS}\left(\mathbf{x}_{\mathrm{R}}^k\right)$ is used, in order to synchronize the magnetometer measurement time t_{m} with the state estimated at step k. For Jacobian computation of $\mathbf{e}_{\mathrm{m}}$, again the IMU propagation Jacobian is required.

Due to measurement inaccuracy and disturbances on-board the airplane, we used rather conservative values for the magnetometer measurement uncertainties, i.e. 5 μT. Note that the magnetometer-specific states are added to the robot state $\mathbf{x}_{\mathrm{R}}$ at time steps k. The time correlations between successive magnetometer-specific states are expressed with relative error terms in the same manner as the bias states of the accelerometer as part of the IMU error term described in Section 5.5.3. This formalism requires additions to the linearized error term and marginalization of magnetometer-specific states analogously, when used as part of the keyframe-based optimization.

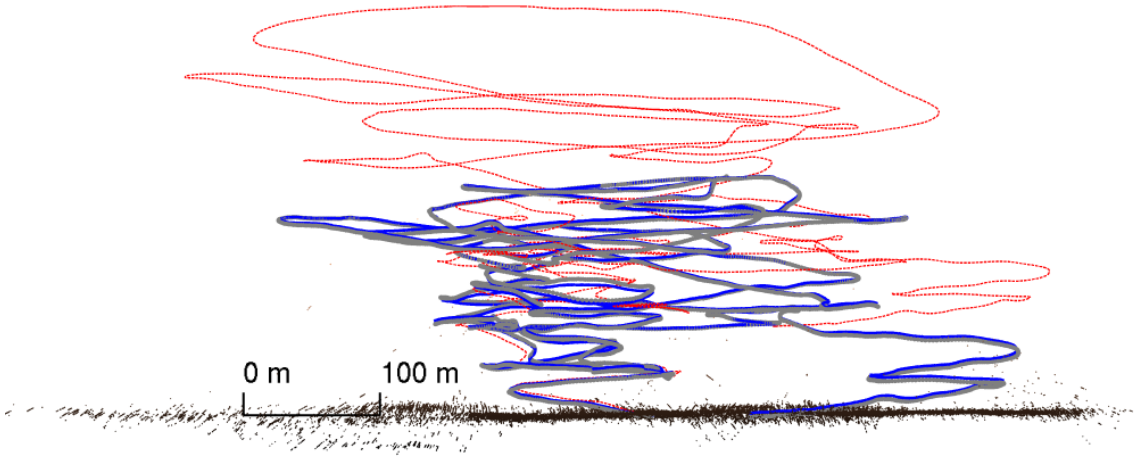
7.3.6 Reconstruction Results

In the following, we will present the results of the landmark and trajectory outputs when offline processing a dataset from a flight lasting 20 minutes, with flight speeds around 10 m/s and heights of over 200 m above ground attained. In Figure 7.20, we show a top and side view of the reconstruction. The output of the algorithm with GPS and without are both displayed. Clearly, the purely monocular visual-inertial odometry (including magnetometer measurements) does not accurately estimate the scale when at higher altitudes above ground, due to weaker observability. This finding underlines the validity of the efforts made to include GPS measurements. Note, however, that for critical parts of the mission, namely during take-off and landing, the state is accurately tracked (relative to the ground) also in absence of GPS.

Analysis of the landmarks as reconstructed with the GPS-based version reveals that the ground is estimated consistently, and expectedly at a higher spatial density whenever the airplane flies close to it. The latter finding is particularly obvious during the approach for landing. The reconstructed local map could thus already be used for preliminary terrain avoidance and automatic landing.



(a) Reconstruction top view (North up).



(b) Reconstruction side view (from South).

Figure 7.20: AtlantikSolar pod reconstruction: the blue path and landmarks were estimated using GPS (gray), whereas the red dashed line corresponds to the algorithm output without GPS. Clearly, the GPS-free version suffers from scale ambiguity at higher altitude.

7.4 Discussion

This chapter overviewed the experience gathered during several years of UAS field testing. Unmanned aircraft, and solar airplanes in particular, are highly complex systems that must operate robustly and long term; therefore, we highly recommend rigorous pre-flight testing, following clearly defined procedures and checklists, as well as establishing introspection and post-processing tools.

Despite the related efforts, a major crash has happened with senseSoar, the causes of which could be identified thanks to logging and analysis tools: in essence, the airplane was most likely stalled in a steep turn close to the ground, leading to a spin.

On the scientific side of flight test results, a focus was given to estimation algorithm outputs. The INS filter not relying on imagery was presented in Section 5.4 and verified with manned glider sensor data. Here, it was analyzed for robustness when processing measurements of the senseSoar airplane. The estimator, in its online implementation, forms the basis for the autopilot of all ASL's airplanes. In a next step, the deployment of the keyframe based stereo-visual-inertial odometry shown in Section 5.5 on-board the Firefly hexacopter was briefly overviewed along with respective use-cases: it provides a robust and accurate input to the controller in a GPS-denied environment, when flying close to structure. Finally, flight tests were performed with the sensing and processing pod mounted to AtlantikSolar. Since solar airplanes typically operate in an environment where GPS and magnetometers provide useful measurements, we introduced how to integrate them with the mono-visual-inertial estimation framework. The output of related state and landmark reconstructions demonstrates the applicability of the method as a first step towards fully autonomous solar airplane operation close to the terrain. Moreover, the version without GPS measurements showed scale estimation problems at higher altitudes, due to poor observability—underlining the validity of the approach chosen to involve additional sensors, e.g. GPS.

The way for future work in various domains has been paved—from the inclusion of more sensors into the estimation to denser environment reconstruction, planning and flying actual application-related missions. The following Chapter 8 will provide an in-depth analysis of future work related to the achievements reported in this work.

Chapter 8
Conclusion

Many different robotics challenges relevant for autonomous solar airplane operations close to the terrain have been addressed within this thesis. Approaches to design have been sketched that led to successful flight-testing of the senseSoar solar airplane and operation of a modular sensing and processing unit. The platforms were given the capability of running efficient and robust algorithms fusing a plethora of sensing modalities for state estimation and mapping—which includes extraction of keypoints from on-board video. The way is paved for the deployment of small solar airplanes of high autonomy flying in ground proximity to perform disaster response, Search and Rescue (SaR), or mapping missions.

8.1 Results and Contribution

In the following, we summarize the specific main results and contributions that have been made with the common goal of long-term autonomous solar airplane operation.

Design of solar airplanes. Solar airplane design starts with sizing of the main components such that a certain mission-specific target criterion is optimally fulfilled; in this respect, a conceptual design tool was presented. As a key innovation with respect to previous work, the structural mass estimation module does not simply resort to statistics. It performs calculation of aerodynamic load cases considering payload and battery mass and chooses the necessary material thicknesses of a simplified shell or rib wing construction. Moreover, the performance evaluation module is capable of using optimal altitude profiles, in order to exploit potential energy storage.

The tool has enabled the analysis of three different use cases of solar airplanes. First of all, we looked at a High-Altitude Long Endurance (HALE) type of airplane that is supposed to fly in a sustained manner above the cloud layer. We have concluded that such an undertaking is at the edge of being feasible, but only in summer and at higher latitudes—while the flight speed of such an airplane would stay significantly below typical wind speeds, rendering such a project doubtful. Next, the performance of a small-scale solar airplane for operation close to the ground was assessed, with the conclusion that such an airplane may fly in a sustained way at medium to high latitudes in summer. Finally, a feasibility study of a record-flying airplane supposed to cross the Atlantic ocean was performed. The analysis revealed that the undertaking is possible when flying in very sunny Summer conditions with little wind close to the water (from West to East)

around 38° North or South. In order to obtain some safety margin (excess time), a size in region of 8 to 14 m of wingspan was found to be optimal.

This analysis has motivated the realization of the *senseSoar* solar airplane—not aimed at sustained flight, but at fully autonomous flight close to the terrain for disaster response, SaR, or mapping missions. As presented in detail, the airplane co-developed with Leichtwerk AG thus makes an engineering contribution across different disciplines; such as solar technology, aerodynamics and flight mechanics, airplane structures, and optimized sensing and processing components. Designed outside the scope of this work, the *AtlantikSolar* airplane has been built for the purpose of crossing the Atlantik Ocean. A size slightly below what was found optimal was chosen, concretely 5.65 m of wingspan, in order to limit complexity and cost. Its realization underlines the validity of the conceptual design approach. The airplane can furthermore be equipped with a modular sensing and processing pod for advanced on-board and offline state estimation and mapping. These platforms provide a unique combination of long endurance flight and capabilities for sensing and processing targeting autonomous navigation close to the terrain.

Moreover, a complete toolchain of aerodynamics and flight dynamics analysis has been presented, that served as a basis in the design process and simulation in general. A fully parametric model was provided that may be used for model-based controller synthesis and that we employ as part of the estimators presented subsequently.

Fail-safe EKF-based state estimation for airplanes. Robust state estimation that is tolerant to failure of sensors has been identified as a core need for long-term solar airplane operation. Thus we presented our work of tightly-coupled sensor fusion from many sources in an EKF framework. The resulting INS filter takes rate gyro, accelerometer, magnetometer, static and dynamic pressure measurements as well as GPS readings and computes the full state that includes the 3D wind vector. The estimator considers the aerodynamic airplane model, allowing it to track Angle of Attack (AoA) and sideslip. In extensive experiments on a manned glider dataset, we demonstrated the robustness of the estimator with respect to outliers that are detected and rejected, as well as the operation under long-lasting GPS-denied conditions: the filter is still able to track 3D relative airspeed and orientation, which are crucial for flight stabilization. To the best of our knowledge, the tight integration, robustness and redundancy by the listed sensors in combination with the consideration of airplane aerodynamics is novel to the field. The respective implementation is light-weight enough to be deployed on all ASL's fixed-wing microcontroller boards, where the estimator has served as the input to the autopilots; the filter is furthermore deployed on-board a commercial glider avionics system for accurate orientation and wind estimation.

Keyframe-based visual-inertial odometry. A core contribution was presented with a further estimator that complements inertial sensing with visual cues. Such techniques enable highly accurate state estimation with drift only in yaw and position; and they furthermore provide situational awareness in the sense of sparsely reconstructed maps consisting of 3D landmarks. In our approach, reprojection error and temporal error terms are jointly minimized in a nonlinear optimization, fully exploiting the stochastic properties of the underlying sensors. The concept of keyframes is implemented by marginalization, i.e. linearization and variable elimination on a part of the problem. The scheme keeps the optimization window bounded in number of states and error terms, while it may still span

an arbitrary time interval. It furthermore maintains the sparsity, thus enabling real-time operation with the capability of scaling the algorithm to specific needs and available processing power. In an extensive quantitative evaluation, comparisons to a competitive MSCKF-implementation as a filtering-based visual-inertial odometry approach are made. The results clearly demonstrate superior accuracy achieved with our method, while coming at admittedly higher computational cost. The framework we describe in detail can operate with a single camera, but also in a stereo or multi-camera version. Moreover, we demonstrated how camera extrinsics can be estimated on-line, even when initialized with a rather rough guess as available e.g. from a CAD model of the visual-inertial sensor setup.

Efficient image keypoints for real-time applications. Vision-supported localization and mapping as intended for on-board deployment asks for high-speed capabilities of forming visual data associations. With this motivation, an image keypoint detection, description and matching scheme was presented, dubbed BRSIK: Binary Robust Invariant Scalable Keypoints. We target the ability for scale and rotation invariance, if needed, as well as tolerance with respect to some amount of viewpoint or brightness change, in order to offer a fast yet high-quality alternative to existing and proven algorithms like SIFT or SURF. We demonstrated a FAST or Harris score based scale-space corner detector, with the option to enforce uniformity in the image; and we introduced a carefully designed sampling pattern applied to the keypoint neighborhood used for local gradient estimation, in order to determine the keypoint orientation. We also use the properly rotated pattern for performing brightness comparisons assembled into a binary bit-string type of descriptor—as inspired by BRIEF—that can be matched extremely fast on today's processor architectures. In quantitative experiments concerning detection repeatability and descriptor similarity, we underlined the validity of the proposed method when compared to SIFT, SURF, BRIEF and also FREAK keypoints. Timings showed the true advantage of BRISK versus SIFT and SURF, since it runs up to 10 times faster. The code was released under BSD and was later integrated into the OpenCV library; both implementations have been widely used and appreciated by the computer vision and robotics communities.

Experiments with unmanned (solar) aircraft. The availability of unique solar airplane platforms with extended sensing and processing capabilities have enabled a series of flight testing. In particular, we have shown the validity of the chosen approaches throughout this work by applying the presented INS filter to all of ASL's fixed-wing airplanes as a basis for their autopilot; it was furthermore only the availability of logged flight-data which has let us analyze a major crash with the senseSoar airplane, leading to the conclusion that it had been stalled in a steep turn close to ground and entered a spin. Moreover, we have demonstrated on-board operation of the stereo-visual-inertial estimator running on a multicopter with similar computational constraints as on the solar airplanes: again, the estimator feeds the controller, thus enabling a high degree of autonomy of the UAS. Finally, flight experiments with the sensing and processing pod mounted to the AtlantikSolar airplane have enabled a prototype-level state estimation and mapping algorithm based on the mono-visual-inertial estimator that was complemented with GPS and magnetometer measurements: the results mark a first milestone towards long-term and fully autonomous small solar airplane operation in ground proximity.

8.2 Future Research Avenues

Future work specific to the individual topics addressed have been suggested on a per-chapter level; here, we focus on the next steps necessary to be given attention when aiming for long-term solar airplane deployment in actual applications.

Development of solar airplanes with enhanced robustness and ease of use. Despite the efforts undertaken in the work described here, actual applications still ask for enhanced robustness, ease of deployment and safety—also when it comes to certification. Partly, such future activities are suggested to be taken over by industrial partners, and partly they need to be addressed already at the stage of research-related platform development. At the mechanical level, a platform that can be quickly assembled and disassembled, including automatic electrical connections is highly desirable, in order to limit dead-times. Similarly, a certain robustness of the airframe with respect to mechanical exposure during take-off and landing as well as water ingress should be given attention. Robustness at software level has proven to be key for successful operation: in particular the low-level processor firmware should be developed with real-time guarantees despite having to handle asynchronous measurements, requests and logging.

Estimation with more sensors integrated. As part of the paradigm of tightly fusing all available information, the integration of pressure measurements and airplane aerodynamic or even fully dynamic models into the visual-inertial estimation process (in addition to magnetometer and GPS) is of highest interest: the resulting framework is expected to estimate wind, Angle of Attack (AoA) and sideslip very accurately while being robust with respect to failure of at least one of the sensor sources. Furthermore, involving thermal imagery in the estimation process will enable a new level of autonomous navigation; in essence, the airplane could fly close to the terrain day and night. Obviously, thermal maps and thermal people detection will form a key element for many of the suggested applications. As a further sensory extension, we suggest to use the wingspan as the (semi-rigid) baseline for stereo vision: the integration of more cameras would enable the computation of depth images in a straight-forward fashion, and further improve accuracy of the estimation—most notably in relation with the scale observability under GPS outage.

Map consistency and multi-agent mapping. The availability of a place recognition and loop-closure module for consistent mapping and localization therein will be of high importance for successive dense mapping steps; furthermore, the promising use-case of collaborative mapping by a multitude of agents will need accurate relative localization for safety and to enable generating a consistent partially overlapping map.

Denser mapping. Safe operation close to the ground, and the execution of autonomous landing in a terrain which may be previously unknown requires the availability of denser environment reconstruction. As a first step, keyframes may be used for multi-view dense stereo using the current state estimator output as initialization. As a second option, inspired by recent advances in computer vision, the approach of optimizing the photometric error and choosing a dense representation of the simultaneously estimated

map could be employed as part of the overall optimization; while the idea seems straightforward, the main challenge will be to achieve real-time performance as well as to find scalable and consistent map representations.

Motion planning along terrain and sense-and-avoid. With the availability of maps, motion planning needs to be addressed, in order to fly scanning missions. While the constraints from the terrain are static, differential constraints of the platform are important to be considered, which in turn are subjected to a dynamically changing environment (i.e. the wind). Furthermore, the capability for sense-and-avoid maneuvers both with respect to static and dynamic objects appearing—such as manned or unmanned aircraft—has been identified as a major requirement when it comes to insertion of UAS into civil airspace.

Robust and fail-safe autopilots. With the availability of parametric dynamic airplane models, employing model-based control strategies such as Linear Quadratic Regulators (LQR), Nonlinear Dynamic Inversion (NDI) or Model Predictive Control (MPC) is highly encouraged: such methodologies come with performance guarantees, allowing to further improve overall robustness, safety as well as power losses due to actuation. Moreover, in addition to the robustness with respect to sensor failures addressed so far, detecting actuator failure, as it happens relatively often with today's RC servos, is of equal importance, along with adaptation of the control strategy such that safe flight is still possible. Moreover, a supervisory level monitoring the system health and deciding on related actions would be desired. Again, this goes hand-in-hand with requirements from civil aviation authorities.

List of Figures

1.1 The senseSoar solar airplane prototype in flight. 1
1.2 UAS design and operation challenges. 2
1.3 Two different unmanned solar airplanes capable of sustained flight. 5
1.4 Landmark measurements. 6

2.1 Some Examples of Solar Airplanes. 10
2.2 Solar power, required power, battery capacity, and performance measures. . 12
2.3 Conceptual design tool operation. 13
2.4 Tennekes' Great Flight Diagram augmented. 16
2.5 Two structure variants of the conceptual design tool. 17
2.6 Altitude, battery energy and irradiation profile. 21
2.7 Maximum constant altitude reachable in sustained flight in summer. 22
2.8 Performance and optimal battery mass of small solar airplane. 24
2.9 Record solar airplane design space. 25
2.10 Atlantic crossing feasibility as a function of latitude and season. 25

3.1 senseSoar and AtlantikSolar airplanes. 30
3.2 senseSoar top, front and side views (CAD). 32
3.3 senseSoar components highlevel overview. 33
3.4 The two solar airplane prototypes. 33
3.5 senseSoar wing shells lying in the negatives. 35
3.6 senseSoar solar modules layout. 36
3.7 senseSoar propulsion units. 36
3.8 senseSoar pod overview. 37
3.9 senseSoar pod assembly and integration. 37
3.10 The AtlantikSolar airplane. 38
3.11 Modular sensing and processing unit. 39

4.1 Simplified representation of the wing vortex system. 42
4.2 Wake Vortex Study by NASA at Wallops Island 42
4.3 2D airfoil flow and forces. 44
4.4 Example lift, drag and moment polars. 45
4.5 Airplane geometry and force definitions. 46
4.6 Horseshoe vortex wing layout. 47
4.7 BEMT geometry cross-section side and front view. 49
4.8 BEMT blade cross-section. 50
4.9 Linearized longitudinal and lateral airplane dynamics. 52

4.10 senseSoar lift, drag and pitch moment polars. 54
4.11 senseSoar glide ratio and climb factor. 54
4.12 Longitudinal poles of senseSoar and physical interpretation of the modes. . . 55
4.13 Lateral poles of senseSoar and physical interpretation of the modes. 55

5.1 Platforms used for filtering. 69
5.2 Overview of filter operation. 70
5.3 Airplane geometry definitions, forces, and moments. 73
5.4 Glider pilot's view. 76
5.5 INS filter trajectory output overhead plot. 78
5.6 Orientation estimates and uncertainty in regular operation. 79
5.7 Accelerometer and gyro bias estimates in reguar operation. 79
5.8 Estimates of altitude and QFF in regular operation. 80
5.9 Wind estimate in regular operation. 80
5.10 Differences of orientation estimates during GPS outage. 81
5.11 ETH main building indoor reconstruction 83
5.12 Coordinate frames of visual-inertial setup. 84
5.13 Full-batch visual inertial factor graph. 85
5.14 Camera and IMU rates. 87
5.15 Visualization of typical data association. 89
5.16 Frames for matching and optimization. 90
5.17 Initial Marginalization 92
5.18 Non-keyframe marginalization 92
5.19 Keyframe marginalization. 93
5.20 Example sparsity pattern. 94
5.21 Visual-inertial sensor front and side view. 95
5.22 Vicon dataset trajectory. 98
5.23 Vicon dataset error statistics. 99
5.24 Bias estimates. 100
5.25 Simon ready for bycicle dataset collection. 100
5.26 Trajectories of the Bicycle Trajectory dataset. 101
5.27 Bicycle dataset relative position error statistics. 101
5.28 ETH dataset trajectories. 102
5.29 ETH dataset error statistics. 103
5.30 Extrinsics calibration trajectories. 104
5.31 extrinsics calibration error statistics. 105
5.32 Convergence of online extrinsics estimation. 105
5.33 Comparison of different frame number settings. 106
5.34 Comparison of different keypoint detection thresholds. 107

6.1 BRISK matching example. 110
6.2 Scale-space interest point detection. 113
6.3 BRISK detection example. 114
6.4 Harris-based detection with uniformity enforcement. 115
6.5 BRISK sampling pattern. 116
6.6 Datasets used for keypoint evaluation. 119
6.7 Detector repeatability scores. 120
6.8 BRISK, SURF and SIFT descriptor similarity. 121

6.9 Comparison of different BRISK2-versions to 64 byte BRIEF. 122
6.10 FREAK versus BRISK2 (with pattern scaled to match FREAK). 123

7.1 senseSoar operations on the airfield of Hausen am Albis. 127
7.2 AtlantikSolar test site. 129
7.3 senseSoar path before crash. 130
7.4 senseSoar crash online state estimates. 131
7.5 senseSoar take-off and flight on Hausen am Albis airfield. 132
7.6 Google Earth overhead plot of the path in normal operation. 133
7.7 Orientation estimates with 3σ-bounds in normal operation. 134
7.8 Remaining estimated states in normal operation. 134
7.9 Angle of Attack and sideslip angles with 3σ-bounds in normal operation. . . 135
7.10 Corrupted data: pressures and magnetometer. 136
7.11 Orientation estimates difference in presence of outliers. 136
7.12 Difference of orientation estimates under GPS outage. 137
7.13 Differences of Angle of Attack and sideslip under GPS outage. 137
7.14 Google Earth overhead plot of the path under simulated GPS outage. 138
7.15 Hexacopter with stereo-visual-inertial sensor. 138
7.16 Hexacopter operating with on-board visual-inertial odometry. 139
7.17 Hexacopter inside a gold mine. 139
7.18 Sensing and processing pod mounted to AtlantikSolar. 140
7.19 AtlantikSolar sensor pod images and keypoint matching. 141
7.20 AtlantikSolar pod reconstruction. 143

List of Tables

2.1 Main criteria determining the airplane structure. 17
2.2 Different structure mass estimation for some examples 26

3.1 senseSoar characteristics. 34
3.2 AtlantikSolar characteristics. 38

5.1 Triadis ISU characteristics. 77
5.2 ADIS16448 IMU characteristics 96
5.3 Dataset characteristics. 97

6.1 Keypoint detection and descriptor extraction timings. 124
6.2 Descriptor matching timings 124

7.1 senseSoar sensor characteristics. 133

Bibliography

European aviation safety agency. cs-22, certification specifications for sailplanes and powered sailplanes. Available at www.easa.eu.int, accessed January 2010, November 2003. 14

Quinetiq newsroom, July 2006. www.qinetiq.com accessed January 2010. 5, 10

Solar impulse news, December 2009. www.solarimpulse.ch, accessed January 2010. 5, 9

STANAG 4671 - unmanned aerial vehicles systems airworthiness requirements (usar). Nato Standardization Agency, 2009. 14

M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: Center surround extremas for realtime feature detection and matching. 2008. 111

A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. IEEE, 2012. 123

T. Barfoot, J. R. Forbes, and P. T. Furgale. Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1-2):101–112, 2011. 65

H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. 2006. 122

H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. 110 (3):346–359, 2008. 7, 111, 113, 118

D. S. Bayard and P. B. Brugarolas. An estimation algorithm for vision-based exploration of small bodies in space. In *American Control Conference, 2005. Proceedings of the 2005*, pages 4589–4595. IEEE, 2005. 60

P. Berry. The sunriser - a design study in solar powered flight. In *Proc. of the World Aviation Conference*, San Diego, USA, Oct 10-12 2000. 15

R. J. Boucher. Project sunrise - a flight demonstration of astro flight model 7404 solar powered remotely piloted vehicle. In *15th Joint Propulsion Conference, AIAA-79-1264, Las Vegas, Nevada, June 18-20*, 1976. 4, 9

S. A. Brandt and F. T. Gilliam. Design analysis methodology for solar-powered aircraft. *AIAA Journal of Aircraft*, 32(4), 1995. 15

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2037–2042. IEEE, 2009. 61

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. 2010. 7, 110, 111, 116, 118

L. Chai, W. A. Hoff, and T. Vincent. Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *Presence: Teleoperators and Virtual Environments*, 11(5):474–492, 2002. 60, 82

M. Chli and A. J. Davison. Active Matching. 2008. 110

J. Crassidis. Sigma-point kalman filtering for integrated gps and inertial navigation. In *Proc. of AIAA Guidance, Navigation, and Control Conference*, San Francisco, CA, August 2005. 58, 59

J. Crassidis, F. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007. 59

A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410 vol.2, Oct 2003. doi: 10.1109/ICCV.2003.1238654. 6, 60

A. J. Davison, N. D. Molton, I. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. 29(6):1052–1067, 2007. 110

T.-C. Dong-Si and A. I. Mourikis. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5655–5662. IEEE, 2011. 6, 61, 91

T.-C. Dong-Si and A. I. Mourikis. Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1064–1071. IEEE, 2012. 61

G. Ducard. *Fault-tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles*. Advances in Industrial Control. Springer, 2009. ISBN 9781848825611. 5, 44, 48, 52

J. A. Duffie and W. A. Beckman. Solar engineering of thermal processes. *NASA STI/Recon Technical Report A*, 81:16591–+, 1980. 19

S. Ebcin and M. Veth. Tightly-coupled image-aided inertial navigation using the unscented kalman filter. Technical report, DTIC Document, 2007. 60

B. Etkin. *Dynamics of atmospheric flight*. Wiley, 1972. ISBN 9780471246206. 5, 44

M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing UAV. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 340 –345, sept. 2008. doi: 10.1109/IROS.2008.4650766. 59

P. Furgale. *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD thesis, University of Toronto, Institute for Aerospace Studies, 2011. 58, 86

P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1280–1286, Nov 2013. doi: 10.1109/IROS.2013.6696514. 96

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 96

D. W. Hall and S. A. Hall. Structural sizing of a solar powered airplane. Contractor Report 172313, NASA, 1984. 15

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988. 88, 110, 112

H. Hertel. *Leichtbau*. Springer-Verlag, 1960. 17

J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2012a. 61, 62

J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Observability-constrained vision-aided inertial navigation. *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep*, 1, 2012b. 62

J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Algorithmic Foundations of Robotics X*, pages 559–574. Springer, 2013. 61, 62, 82

G. Huang, M. Kaess, and J. J. Leonard. Towards consistent visual-inertial navigation. *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep*, 2013. 61, 62

G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. A first-estimates jacobian ekf for improving slam consistency. In *Experimental Robotics*, pages 373–382. Springer, 2009. 62

G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. An observability-constrained sliding window filter for slam. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 65–72. IEEE, 2011. 6, 61, 62

V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Information Fusion (FUSION), International Conference on*, 2012. 60, 61, 82

F. G. Irving and D. Morgan. The feasibility of an aircraft propelled by solar energy. In *Proc. of the 2nd International Symposium on the Technology and Science of Low Speed and Motorless Flight*, Cambridge, Massachusetts, September 11-13 1974. AIAA-1974-1042. 10

J. S. Jang and D. Liccardo. Automation of small UAVs using a low cost mems sensor and embedded computing platform. In *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, pages 1 –9, oct. 2006. doi: 10.1109/DASC.2006.313772. 59

C. Jia and B. L. Evans. Probabilistic 3-d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *MMSP*, pages 203–208, 2012. 61

E. Johnson, S. Fontaine, and A. Kahn. Minimum complexity uninhabited air vehicle guidance and flight control system. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 3A4/1–3A4/9 vol.1, Oct 2001. doi: 10.1109/DASC.2001.963345. 59

E. S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *International Journal of Robotics Research (IJRR)*, 30(4):407–430, 2011. 60, 61

S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. pages 182–193, 1997. 59

S.-H. Jung and C. J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–732. IEEE, 2001. 60, 82

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012. 61

R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 58

Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. 2004. 111

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. 60, 61

J. Kim and S. Sukkarieh. Real-time implementation of airborne inertial-slam. *Robotics and Autonomous Systems*, 55(1):62–71, 2007. 60

G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007. 60

L. Kneip and P. T. Furgale. OpenGV: A unified and generalized approach to real-time calibrated geometric vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 89

K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *Robotics Research*, pages 201–212. Springer, 2011a. 60

K. Konolige, M. Agrawal, and J. Solá. Large-scale visual odometry for rough terrain. *Robotics Research*, 66:201–212, 2011b. 96, 103

D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2012. 62

M. Kumon, I. Mizumoto, Z. Iwai, and M. Nagata. Wind estimation by unmanned air vehicle with delta wing. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1896–1901, April 2005. doi: 10.1109/ROBOT.2005.1570390. 59

J. W. Langelaan, N. Alley, and J. Neidhoefer. Wind field estimation for small unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4):1016–1030, 2011. 59

G. J. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2006. 48

S. Leutenegger and R. Siegwart. A low-cost and fail-safe inertial navigation system for airplanes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 612–618, May 2012. doi: 10.1109/ICRA.2012.6225061. 59, 72

S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 88, 116

S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2013. 61, 63, 82, 94

M. Li and A. I. Mourikis. Improving the accuracy of ekf-based visual-inertial odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012a. 61, 82

M. Li and A. I. Mourikis. Improving the accuracy of ekf-based visual-inertial odometry. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 828–835. IEEE, 2012b. 60, 61, 82

M. Li and A. I. Mourikis. Vision-aided inertial navigation for resource-constrained systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1057–1063. IEEE, 2012c. 60

M. Li and A. I. Mourikis. Optimization-based estimator design for vision-aided inertial navigation. *Robotics*, page 241, 2013. 60

M. Li, B. H. Kim, and A. I. Mourikis. Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4712–4719. IEEE, 2013. 60, 61, 82

K. Lievens, J. Mulder, and P. Chu. Single gps antenna attitude determination of a fixed wing aircraft aided with aircraft aerodynamics. In *AIAA Guidance, Navigation, and Control Conference and Exhibit. San Francisco, California*, 2005. 59

J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575, 2007. 61

D. G. Lowe. Distinctive image features from scale-invariant keypoints. 60(2):91–110, 2004. 7, 110

S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3923–3929. IEEE, 2013. 60

E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. 2010. 110, 112, 118

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, 2011. 61

A. Martinelli. Visual-inertial structure from motion: observability vs minimum number of sensors. *ICRA 2014*, 2014. 61

B. McCormick. *Aerodynamics, aeronautics, and flight mechanics*. Wiley, 1979. ISBN 9780471030324. 5, 41, 43, 44

C. Mei, G. Sibley, M. Cummins, P. M. Newman, and I. D. Reid. Rslam: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 198–214, 2011. 63, 83

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. 2003. 111

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. 2:1115–1125, 2005. 111, 119

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool. A comparison of affine region detectors. 65(1):43–72, 2005. ISSN 0920-5691. 110, 111, 119

F. M. Mirzaei and S. I. Roumeliotis. IMU-camera calibration: Bundle adjustment implementation. Technical report, Department of Computer Science and Engineering, Univerisy of Minnesota, 2007. 61

F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156, 2008. 61

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 (ICRA)*, 2007. 6, 60, 82, 87

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. *Robotics, IEEE Transactions on*, 25(2):264–280, 2009. 63, 82, 84, 95

E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis. C-klam: Constrained keyframe-based localization and mapping. In *In Proc. of the Workshop on "Multi-View Geometry in Robotics" at the Robotics: Science and Systems (RSS)*, 2013. 61, 82

J. Nikolic, J. Rehder, P. G. Michael Burri, S. Leutenegger, P. T. Furgale, and R. Siegwart. A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014. 39, 95

T. E. Noll, J. M. Brown, M. E. Perez-Davis, S. D. Ishmael, G. C. Tiffany, and M. Gaier. Investigation of the helios prototype aircraft mishap. Technical report, NASA Langley Research Center, 2004. 3, 10

A. Noth. History of solar flight. Technical report, ETH Zurich, 2006. 10, 29

A. Noth. *Design of Solar Powered Airplanes for Continuous Flight*. PhD thesis, Autonomous Systems Lab, ETH ZÃ$\frac{1}{4}$rich, Switzerland, December 2008. 4, 5, 10, 14, 15, 16, 19, 29, 54

W. H. Philips. Some design considerations for solar-powered aircraft. Technical Report 1675, NASA, 1980. 10, 19

A. Ranganathan, M. Kaess, and F. Dellaert. Fast 3d pose estimation with out-of-sequence measurements. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007. 60

M. A. Rehmet, R. Voit-Nitschmann, and B. Kröplin. Eine methode zur auslegung von solarflugzeugen. Technical Report DGLR-JT97-031, Universität Stuttgart, Germany, 1997. 15

P. J. Roberts, R. A. Walker, and P. O'Shea. Fixed wing uav navigation and control through integrated gnss and vision. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–13, 2005. 59

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. 2006. 7, 110

S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4326–4333. IEEE, 2002. 60, 82

O. Schrenk. *A simple approximation method for obtaining the spanwise lift distribution*. National Advisory Commitee for Aeronautics, 1940. 14

E.-H. Shin. *Estimation techniques for low-cost inertial navigation*. PhD thesis, University of Calgary (Canada), 2005. available from http://www.proquest.com (publication number AAT NR06960). 59

E.-H. Shin and N. El-Sheimy. An unscented kalman filter for in-motion alignment of low-cost imus. In *Position Location and Navigation Symposium, 2004. PLANS 2004*, pages 273–279. IEEE, 2004. 58, 60

G. Sibley, L. Matthies, and G. Sukhatme. Sliding window filter with application to planetary landing. *Journal of Field Robotics*, 27(5):587–608, 2010. 6, 61

J. C. Sivells and R. H. Neely. Method for calculating wing characteristics by lifting-line theory using nonlinear section lift data. Technical report, DTIC Document, 1947. 43

H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010. 6, 60

D. Strelow and S. Singh. Online motion estimation from image and inertial measurements. 2003. 60

D. Strelow and S. Singh. Motion estimation from image and inertial measurements. *The International Journal of Robotics Research*, 23(12):1157–1195, 2004. 60

H. Tennekes. The simple science of flight - from insects to jumbo jets. MIT Press, Cambridge, Massachusetts, USA, 1992. 15

S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research (IJRR)*, 25 (5):403–430, 2006. 86

N. F. Toda, J. L. Heiss, and F. H. Schlee. Spars: The system, algorithm, and test results. Technical Report TR-0066(5306)-12, IBM Federal Systems Division, Owego, New York, 1969. pp. 361-370. 58

E. Tola, V. Lepetit, and P. Fua. Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. 32(5):815–830, 2010. 116

C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991. 7

N. Trawny and S. I. Roumeliotis. Indirect kalman filter for 3d attitude estimation. Technical Report 2005-002, Rev. 57, University of Minnesota, Dept. of Computer Science and Engineering, 4-192 EE/CS Building, 200 Union St. S.E., Minneapolis, MN 55455, January 2005. 58

R. van der Merwe and E. Wan. Sigma-point kalman filters for integrated navigation. In *Proceedings of the 60th Annual Meeting of The Institute of Navigation (ION)*, Dayton, OH, June 2004. URL `http://cslu.cse.ogi.edu/publications/ps/merwe04a.pdf`. 59

S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012. 60, 61, 82

S. M. Weiss. *Vision based navigation for micro helicopters*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 20305, 2012, 2012. 61

About the Author

Stefan Leutenegger was born on February 16th 1983 in Winterthur, Switzerland. He received his B.Sc. in Mechanical Engineering from ETH Zurich in 2006 and his M.Sc. (with honors) in Robotics likewise from ETH in 2008. During his studies, he worked as teaching and research assistant for various courses and institutes. Since 2009, he has been a PhD student at ASL.

Stefan has worked on solar airplane design from concepts to realization and flight testing, as well as related algorithms for navigation close to the terrain. His activities covered a broad range from structural, electrical and software engineering to the development of highly efficient, robust, and accurate algorithms for multi-sensor state estimation and mapping.

As part of his PhD work, Stefan spent three months at the robotics company Willow Garage in Menlo Park, California, in 2012 under the supervision of Dr. Kurt Konolige and Dr. Vincent Rabaud.

Besides his involvement in engineering and science activities, Stefan has had the ASL-internal lead in the European FP7 Projects "ICARUS" and "SHERPA" since the proposal writing phase. He has furthermore supervised dozens of student projects at Bachelor and Master level. Moreover, he was one of the main supervisors of the Focus Project "Skye" realized by a group of 16 Bachelor students during the course of one year. Stefan has also been teaching the fixed-wing part of ASL's Master course on Unmanned Aerial Systems.

In his private life, Stefan Leutenegger is a passionate glider pilot enjoying the challenges of long cross-country flying as well as (international) competitions.

List of publications

Book chapters

- S. Leutenegger, C. Huerzeler, A. K. Stowers, K. Alexis, M. Achtelik, D. Lentink, P. Oh, and R. Siegwart. Flying Robots, *Springer Handbook of Robotics*, second edition. B. Siciliano and O. Khatib, editors. To appear in 2015.

Peer-reviewed journals

- S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization. In *The International Journal of Robotics Research (**IJRR**)*, submitted.
- S. Leutenegger, M. Jabas, and R. Siegwart. Solar Airplane Conceptual Design and Performance Estimation. In *Journal of Intelligent and Robotic Systems (**JINT**)*, Vol. 61, No. 1-4, 2011.

Peer-reviewed conference proceedings

- S. Leutenegger, A. Melzer, K. Alexis, and R. Siegwart. Robust State Estimation for Small Unmanned Airplanes. To appear in *Proceedings of the IEEE International Multi-Conference on Systems and Control (**MSC**)*, 2014.
- J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart. A Synchronized Visual-Inertial Sensor System with FPGA Pre-Processing for Accurate Real-Time SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (**ICRA**)*, 2014.
- M. Burri, et al. Design and Control of a Spherical Omnidirectional Blimp. In *Proceedings of the IEEE International Confefence on Intelligent Robots and Systems (**IROS**)*, 2013.
- S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige and R. Siegwart. Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. In *Proceedings of Robotics: Science and Systems (**RSS**)*, 2013.
- J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart. A UAV system for inspection of industrial facilities. In *Proceedings of the IEEE Aerospace Conference (**AeroConf**)*, 2013.
- M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart. State Estimation for Legged Robots- Consistent Fusion of Leg Kinematics and IMU. In *Robotics: Science and Systems (**RSS**)*, 2012.
- S. Leutenegger, and R. Siegwart. A Low-Cost and Fail-Safe Inertial Navigation System for Airplanes. In *Proceedings of the IEEE International Conference on Robotics and Automation (**ICRA**)*, 2012.
- S. Leutenegger, M. Chli and R. Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (**ICCV**)*, 2011. Cited over 300 times (as of May 2014).

- P. Fankhauser, S. Bouabdallah, S. Leutenegger, and R. Siegwart. Modeling and decoupling control of the coax micro helicopter. In *Proceedings of the IEEE International Confefence on Intelligent Robots and Systems (**IROS**)*, 2011.
- C. Bermes, S. Leutenegger, S. Bouabdallah, D. Schafroth, R. Siegwart. New design of the steering mechanism for a mini coaxial helicopter. In *Proceedings of the IEEE International Confefence on Intelligent Robots and Systems (**IROS**)*, 2008.
- D. J. Bell, S. Leutenegger, K. M. Hammar, L. X. Dong, and B. J. Nelson. Flagella-like propulsion for microrobots using a nanocoil and a rotating electromagnetic field. In *Proceedings of the IEEE International Conference on Robotics and Automation (**ICRA**)*, 2007.

Other publications

- A. Noth, S. Leutenegger, R. Y. Siegwart, and W. Engel. Designing solar airplanes for continuous flight. *SPIE Newsroom on Solar & Alternative Energy*, 2009.

Thesis

- S. Leutenegger. End Effector Design and Control of 5 DoF Planetary Rover Robot Arm for Communication Relay Retrieval, *Master Thesis*, 2008, ETH Zurich and NASA Ames Research Center. Supervised by Prof. R. Siegwart, Dr. T. Fong and Dr. V. SunSpiral.
- S. Leutenegger. Mechanical Design of a CoG Shifting Steering Mechanism for a Coaxial Autonomous Helicopter, *Semester Thesis*, 2007, ETH Zurich. Supervised by Prof. R. Siegwart and Dr. C. Bermes.
- S. Leutenegger. Towards a Microswimmer Propulsion System, *Bachelor Thesis*, 2006, ETH Zurich. Supervised by Prof. B. C. Nelson and Dr. D. Bell

Patents

- S. Leutenegger, et al. Omnidirectional Spherical Blimp, 2012, Swiss patent (provisional).